

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



PROYECTO FIN DE CARRERA

**Implementación y evaluación de algoritmos de cálculo de ruta y
selección de longitud de onda para un PCE multicapa**

Sergio Martínez de Tejada Ayuso

JULIO 2013

Implementación y evaluación de algoritmos de cálculo de ruta y selección de longitud de onda para un PCE multicapa

AUTOR: Sergio Martínez de Tejada Ayuso

TUTOR: Víctor López Álvarez

PONENTE: Jorge E. López de Vergara Méndez

Dpto. Tecnología Electrónica y de las Comunicaciones

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Julio de 2013

Resumen

En los últimos años los servicios IP punto a punto han aumentado considerablemente. Esto obliga a desarrollar múltiples técnicas de ingeniería de transporte. Una de estas técnicas es la arquitectura *Generalized Multi-Protocol Label Switching* (GMPLS), que surge como modelo para que las redes de próxima generación, formadas por elementos como routers, switches, *Dense Wavelength Division Multiplexing* (DWDM) *Systems*, *Add-Drop Multiplexors* (ADMs), *Photonic Cross-Connects* (PXCs), *Optical Cross-Connects* (OXCs), etc. puedan proveer recursos dinámicamente a la red y asegurar la estabilidad de ésta mediante técnicas de protección y restauración.

Se ha probado que GMPLS se puede aplicar para el control redes multicapa, puesto que permite eficientemente dar servicios punto a punto gracias a que puede soportar múltiples tipos de conmutación. Este modelo permite además a los nodos solicitar caminos punto a punto de una manera distribuida, enviando peticiones al PCE. El PCE es el módulo encargado de procesar todas las peticiones de conexión de la red, calculando rutas y asignándolas a cada petición. El problema es que cuando tenemos redes multicapa este cálculo de caminos no es tan sencillo y puede llegar a ser computacionalmente costoso. Por ello se han desarrollado diversas implementaciones de algoritmos de encaminamiento para intentar reducir la complejidad de los cálculos de rutas y conseguir así reducir los costes que conlleva este proceso. También distintas arquitecturas de PCE multicapa pueden simplificar este problema. Al tener distintas capas con distintos recursos, podemos tener un PCE que compute caminos en cada una de ellas o tener un PCE completo multicapa capaz de computar caminos punto a punto pasando por distintas capas y haciendo uso de múltiples tecnologías de computación. Éste último caso tiene las garantías de ser el más eficiente a la hora de aprovechar los recursos de los que disponemos en nuestra red.

Palabras Clave

PCE, cálculo de caminos, multicapa, algoritmos, encaminamiento, asignación de longitud de onda, probabilidad de bloqueo

Abstract

In the last years IP services have increased significantly. This involves developing multiple transport engineering techniques for different networks with diverse switching capabilities. One of these techniques is the Generalized Multi-Protocol Label Switching (GMPLS) architecture, which arises as a model for next-generation networks. These networks consist of elements such as routers, switches, Dense Wavelength Division Multiplexing (DWDM) Systems, add-drop multiplexers (ADMs), Photonic Cross-Connects (PXC), Optical Cross-Connects (OXC), etc. These elements and GMPLS provide network resources dynamically and ensure the stability through protection and restoration techniques.

GMPLS can be used to the control plane in multi-layer networks to provide efficiently point-to-point services. This model also allows nodes to request peer-to-peer paths in a distributed manner, sending requests to the PCE. PCE is the module responsible for processing all these connections in the network, by computing routes and assigning resources and paths to each request. The problem appears when we have multilayer networks, in these cases the path computation is not so simple and can be computationally expensive. To solve this problem, we have developed several implementations of routing algorithms that attempt to reduce the complexity of the operations to achieve lower costs associated with this process. Having different layers with diverse resources lead us to two different PCE architectures: one PCE for each layer which can compute single-layer paths or having a multilayer PCE, which can compute paths through different layers and using multiple switching technologies. The latter case has the guarantees to be the most efficient in harnessing the resources that we have in the whole network.

Key Words

PCE, path computation, multilayer, algorithms, routing, wavelength assignment, blocking probability

Agradecimientos

En primer lugar quiero agradecer a toda mi familia por el esfuerzo que han puesto siempre en darme una educación y apoyarme siempre en todo lo que he hecho o intentado hacer durante mi carrera. Gracias a mis padres por darme la posibilidad de haber hecho casi todo lo que he querido durante estos años y por confiar en mi responsabilidad. Gracias también a mis hermanos Marina y Miguel por aguantarme durante tanto tiempo, sé que soy un poco pesado. También a mi abuela por todo el cariño, le hará mucha ilusión que termine la carrera.

En segundo lugar quería dar las gracias a todos mis amigos de siempre, tres de los cuáles han sido compañeros de Universidad (Carlos, Tomás y Vicente), y a todos los compañeros y amigos con los que durante toda la carrera he compartido momentos muy buenos y momentos de saturación, aunque al final de todo solo quedan los buenos. Ha sido un verdadero placer pasar todos estos años juntos.

Quiero agradecer a mis actuales compañeros de piso (Carlos, Marta, Eider e Iñigo) el buen “rollo” que mantenemos en la casa, esto favorece mucho en estos momentos de trabajo duro.

También recuerdo con añoranza en estos momentos el año que pase de Erasmus en Roma. Por ello quiero agradecer también a toda la gente que conocí, y en especial a los buenos amigos y amigas que hice, por un año increíble y lleno de recuerdos. Gracias a todos.

Y por último agradecer a mi tutor, Víctor López, por darme la posibilidad de realizar este proyecto, mostrándome así infinidad de cosas en el mundo de las redes ópticas. La verdad que al principio de todo esto ni me imaginaba que podría llegar a aprender tantas cosas. También agradecer a la empresa Telefónica I+D por darme la posibilidad de trabajar como becario durante este último año con toda disponibilidad de recursos. Gracias a todos mis compañeros de trabajo, de nuevo mi tutor Víctor entre ellos, y en especial a mis compañeros becarios, Marta y Arturo. Es un placer poder trabajar con todos vosotros y espero que lo siga siendo durante mucho tiempo.

A todos, gracias.

INDICE DE CONTENIDOS

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	2
1.3	Estructura de esta memoria	2
1.4	Medios a utilizar	3
2	Modelo de Red de nueva generación	5
2.1	Arquitectura de red de próxima generación	6
2.2	Plano de control: GMPLS	6
2.2.1	OSPF	10
2.2.2	RSVP	11
2.2.3	LMP	11
2.3	Plano de Control Multicapa	13
2.3.1	Modelos de plano de control multicapa	15
2.3.1.1	Modelo Overlay	15
2.3.1.2	Modelo Peer: Modelo integrado	17
2.3.1.3	Border Peer Model	18
2.3.2	Requisitos y funcionalidades del plano de control integrado	19
2.4	Path Computation Element	20
2.4.1	Arquitectura general	20
2.4.2	Modelos de funcionamiento del PCE en entornos multicapa	23
2.5	VNTM	24
3	Definición de los escenarios de experimentación	27
3.1	Testbed de Adrenaline	27
3.1.1	Descripción funcional	27
3.1.2	Descripción del plano de control	28
3.2	Testbed de Telefónica I+D	30
3.2.1	Descripción funcional	30
3.2.2	Descripción del plano de control	32
4	Evaluación de algoritmos para redes de conmutación de longitudes de onda	39
4.1	Definición de algoritmos implementados	39
4.1.1	KSP con Pre-cómputo (<i>Fixed Routing</i>)	39
4.1.2	AURE	39
4.1.3	KSP cada conexión (<i>Live</i>)	40
4.2	Criterio de selección de longitud de onda	40
4.2.1	Pack	41
4.2.2	Spread	42
4.2.3	Random	42
4.2.4	Exhaustive	42
4.2.5	Fixed (<i>First Fit</i>)	42
4.3	Efecto del Bloqueo de conexión por “robo de lambda”	43
4.4	Evaluación de los resultados	43
4.4.1	Resultados CTTC	43
4.4.1.1	Probabilidad de bloqueo	43
4.4.1.2	Tiempos de ejecución	44
4.4.2	Resultados TID	47
4.4.2.1	Sin Pre-reserva	48
4.4.2.2	Con Pre-reserva	50
5	Evaluación de algoritmos para redes multicapa	53

5.1	Modelo Separado	53
5.2	Modelo Integrado.....	55
5.2.1	Construcción del Grafo Auxiliar	55
5.2.2	Asignación de pesos al Grafo Auxiliar.....	58
5.3	Implementación y procedimientos de los algoritmos expuestos para cada modelo.....	60
5.3.1	Modelo separado	60
5.3.2	Modelo integrado	61
5.4	Evaluación de los resultados.....	62
5.4.1	Validación de las distintas políticas	62
5.4.2	Comparación de Resultados PCE Multicapa con TEDs separadas y TED multicapa	66
5.4.3	Tiempos de Computación en PCE Multicapa	70
6	Conclusiones y futuras líneas de trabajo.....	72
6.1	Conclusiones.....	72
6.2	Futuras líneas de trabajo	73
	Referencias.....	75
	Glosario.....	79
	Anexos	I
A	Resultados Multicapa NSFNet 14	I
B	Manual del Usuario.....	III
C	Publicaciones Relacionadas	VI
D	PRESUPUESTO.....	X

INDICE DE FIGURAS

FIGURA 1-1: RED DE TRANSPORTE DE PRÓXIMA GENERACIÓN	1
FIGURA 2-1: ESQUEMA MODELO DE RED ASON	5
FIGURA 2-2: MODELO GMPLS BASADO EN RED ÓPTICA.....	7
FIGURA 2-3: JERARQUÍA DE TIPOS DE CONMUTACIÓN.....	8
FIGURA 2-4: TIPOS DE ROUTERS (LSR).....	8
FIGURA 2-5: DIAGRAMA DE FUNCIONALIDAD DEL PLANO DE CONTROL [16]	10
FIGURA 2-6: LINK ENTRE LOS NODOS A Y B.....	11
FIGURA 2-7: INTERCAMBIO DE MENSAJES ENTRE NODOS PARA COMPROBAR LA CONECTIVIDAD DEL LINK	12
FIGURA 2-8: ARQUITECTURA MODELO <i>OVERLAY</i> [30]	16
FIGURA 2-9: ESQUEMA MODELO PEER [30]	18
FIGURA 2-10: MODELO <i>BORDER PEER</i> [30]	19
FIGURA 2-11: DIAGRAMA DE FUNCIONALIDAD PCE	21
FIGURA 2-12: PCEs EN CAPAS SEPARADAS	23
FIGURA 2-13: PCE MULTICAPA	24
FIGURA 2-14: ESQUEMA MODELO DE RED MULTICAPA CON VNTM	25
FIGURA 2-15: ESQUEMA FUNCIONAMIENTO CON VNTM	26
FIGURA 3-1: ARQUITECTURA DEL PCE EN EL <i>TESTBED</i> DE ADRENALINE	28
FIGURA 3-2: EUROPEAN OPTICAL NETWORK	30
FIGURA 3-3: DIAGRAMA DE TIEMPOS DE FUNCIONAMIENTO DEL <i>TESTBED</i>	31
FIGURA 3-4: ESQUEMA MODELO DE PLANO DE CONTROL DE TID	32
FIGURA 3-5: MENSAJES CAPTURADOS CON <i>WIRESHARK</i>	33
FIGURA 3-6: MENSAJE DE PETICIÓN PCEP	33
FIGURA 3-7: RESPUESTA ERO MULTICAPA.....	34
FIGURA 3-8: RESPUESTA ERO IP/MPLS	34
FIGURA 3-9: DIAGRAMA DE INTERCAMBIO DE MENSAJES PARA RESPUESTA ML	35
FIGURA 3-10: DIAGRAMA DE INTERCAMBIO DE MENSAJES EN CASO DE RESPUESTA IP/MPLS	36
FIGURA 3-11: TOPOLOGÍA DE LA NSF NETWORK DE EE.UU.	36
FIGURA 3-12: TOPOLOGÍA RED DE TELEFÓNICA ESPAÑA	37
FIGURA 4-1: ESQUEMA DE FUNCIONAMIENTO DEL AURE	40
FIGURA 4-2: RED ÓPTICA: EJEMPLO DEMOSTRACIÓN SELECCIÓN DE LAMBDA.....	41
FIGURA 4-3: PROBABILIDADES DE BLOQUEO VARIANDO LA CARGA DE LA RED EN EL CTTC	44
FIGURA 4-4: HISTOGRAMA DE TIEMPOS DE COMPUTACIÓN CTTC	45
FIGURA 4-5: FUNCIÓN DE DISTRIBUCIÓN ACUMULADA PARA LOS TIEMPOS DE COMPUTACIÓN EN EL CTTC.....	45
FIGURA 4-6: HISTOGRAMA DE TIEMPOS DE <i>SETUP</i> DE LOS LSPs EN EL CTTC.....	46
FIGURA 4-7: FUNCIÓN DE DISTRIBUCIÓN ACUMULADA PARA LOS TIEMPOS DE <i>SETUP</i> DE LOS LPSS EN EL CTTC.....	47
FIGURA 4-8: PROBABILIDADES DE BLOQUEO AURE PARA DISTINTAS POLÍTICAS DE SELECCIÓN DE LAMBDA.....	48
FIGURA 4-9: COMPARACIÓN DE KSP CON PRE CÓMPUTO, KSP <i>ON LIVE</i> Y AURE PARA UNA SELECCIÓN DE LAMBDA FF	49
FIGURA 4-10: TIEMPOS DE COMPUTACIÓN SIN PRE RESERVA.....	50
FIGURA 4-11: PROBABILIDAD DE BLOQUEO DEL AURE CON DISTINTAS POLÍTICAS DE SELECCIÓN DE LAMBDA CON PRE RESERVA	51
FIGURA 4-12: COMPARACIÓN PROBABILIDAD DE BLOQUEO KSP <i>ON LIVE</i> , KSP <i>PRECOMP</i> Y AURE CON PRE RESERVA Y SELECCIÓN DE LAMBDA FF	52
FIGURA 4-13: TIEMPOS DE COMPUTACIÓN CON PRE RESERVA	52
FIGURA 5-1: TOPOLOGÍA FÍSICA (ÓPTICA) DE LA RED DE EJEMPLO	55
FIGURA 5-2: TOPOLOGÍA VIRTUAL DE EJEMPLO	56
FIGURA 5-3: GRAFO AUXILIAR DE LA RED DE EJEMPLO	56
FIGURA 5-4: DIRECCIÓN IPV4 DE LOS NODOS DEL GRAFO AUXILIAR.....	56

FIGURA 5-5: PROPIEDADES ASOCIADAS A CADA NODO	57
FIGURA 5-6: <i>INFO TABLE</i> PARA LA RELACIÓN ENTRE LOS NODOS AUXILIARES Y SUS PROPIEDADES.....	57
FIGURA 5-7: RELACIONES ENTRE LOS NODOS DEL GRAFO AUXILIAR Y SUS IDENTIFICADORES	57
FIGURA 5-8: ÁRBOLES ORIGEN Y DESTINO EN LA OPERACIÓN 4	61
FIGURA 5-9: COMPARACIÓN DE POLÍTICAS MOSTRANDO UTILIZACIÓN DE RECURSOS EN 2 TEDS..	62
FIGURA 5-10: COMPARACIÓN DE POLÍTICAS MOSTRANDO UTILIZACIÓN DE RECURSOS EN 1 TED..	63
FIGURA 5-11: GRÁFICA DEL NÚMERO DE OPERACIONES EN CADA ALGORITMO PARA EL MODELO DE 1 TED INTEGRADA	64
FIGURA 5-12: RESULTADOS DE PROPORCIÓN DE OPERACIONES USADAS EN CADA ALGORITMO PARA EL MODELO DE 1 TED INTEGRADA	64
FIGURA 5-13: GRÁFICA DEL NÚMERO DE OPERACIONES EN CADA ALGORITMO PARA EL MODELO DE 2 TEDS SEPARADAS	64
FIGURA 5-14: GRÁFICA DEL NÚMERO DE OPERACIONES EN CADA ALGORITMO PARA EL MODELO DE 2 TED INTEGRADA	65
FIGURA 5-15: EVOLUCIÓN NÚMERO DE OPERACIONES 1 PARA LOS DISTINTOS CASOS EN SIMULACIÓN ESTÁTICA CON 5000 CONEXIONES	65
FIGURA 5-16: EVOLUCIÓN NÚMERO DE OPERACIONES 2 PARA LOS DISTINTOS CASOS EN SIMULACIÓN ESTÁTICA CON 5000 CONEXIONES	65
FIGURA 5-17: EVOLUCIÓN NÚMERO DE OPERACIONES 3 PARA LOS DISTINTOS CASOS EN SIMULACIÓN ESTÁTICA CON 5000 CONEXIONES	66
FIGURA 5-18: EVOLUCIÓN NÚMERO DE OPERACIONES 4 PARA LOS DISTINTOS CASOS EN SIMULACIÓN ESTÁTICA CON 5000 CONEXIONES	66
FIGURA 5-19: COMPARACIÓN DE POLÍTICAS. NÚMERO DE CONEXIONES BLOQUEADAS	67
FIGURA 5-20: COMPARACIÓN DE POLÍTICAS. ANCHO DE BANDA TRANSMITIDO	67
FIGURA 5-21: RESULTADOS DE LA OCUPACIÓN DE LAMBDA EN LA RED DE TELEFÓNICA ESPAÑA CON 1 TEDS SEPARADAS	68
FIGURA 5-22: RESULTADOS DE LA OCUPACIÓN DE LAMBDA EN LA RED DE TELEFÓNICA ESPAÑA CON 2 TEDS SEPARADAS	69
FIGURA 5-23: RESULTADOS DEL RENDIMIENTO VARIANDO EL NÚMERO DE LAMBDA EN LA RED NSFNET14 CON 2 TEDS SEPARADAS	69
FIGURA 5-24: RESULTADOS DEL RENDIMIENTO VARIANDO EL NÚMERO DE LAMBDA EN LA RED NSFNET14 CON 1 TED MULTICAPA	70
FIGURA 5-25: RESULTADOS DE LOS TIEMPOS DE COMPUTACIÓN PARA 2 Y 1 TEDS PARA LOS ALGORITMOS MINTH Y MINLP	71
FIGURA A-1: EJEMPLO REGISTRO DEL ALGORITMO EN EL PCESERVERCONFIGURATION.XML	I
FIGURA A-2: EJEMPLO REGISTRO DEL ALGORITMO EN EL PCESERVERCONFIGURATION.XML	II
FIGURA B-3: EJEMPLO REGISTRO DEL ALGORITMO EN EL PCESERVERCONFIGURATION.XML	III
FIGURA B-4: UTILIZACIÓN DEL PUERTO DE GESTIÓN	V

INDICE DE TABLAS

TABLA 4-1: TABLA DE MATRIZ DE UTILIZACIÓN ACTUAL DE CADA LONGITUD DE ONDA EN LA RED	41
TABLA 5-1: TABLA DE COMPARACIÓN DE LAS CUATRO OPERACIONES.....	54
TABLA 5-2: TABLA DE PESOS ASOCIADOS A CADA LINK EN LOS EXPERIMENTOS PARA LAS TRES POLÍTICAS DE ENCAMINAMIENTO.....	59
TABLA 5-3: TABLA CON LAS CONEXIONES EN LAS QUE APARECE EL PRIMER BLOQUEO EN CADA UNA DE LAS REDES Y PARA LAS 2 POLÍTICAS DE ENCAMINAMIENTO	68
TABLA 5-4: TABLA CON LAS CONEXIONES TOTALES ESTABLECIDAS EN CADA UNA DE LAS REDES Y PARA LAS 2 POLÍTICAS DE ENCAMINAMIENTO	68
TABLA C-1: TABLA DE GASTOS EN RECURSOS DE SOFTWARE	X
TABLA C-2: TABLA DE GASTOS EN RECURSOS HARDWARE	X
TABLA C-3: TABLA DE GASTOS EN RECURSOS MATERIALES.....	X
TABLA C-4: TABLA DE PLANIFICACIÓN DEL PROYECTO DESGLOSADO EN TAREAS	XI
TABLA C-5: TABLA DE COSTES SALARIALES	XII
TABLA C-6: TABLA DE COSTES DE MANO DE OBRA.....	XII
TABLA C-7: TABLA DE PRESUPUESTO DE EJECUCIÓN POR CONTRATA	XIII
TABLA C-8: TABLA DE PRESUPUESTO TOTAL	XIII

1 Introducción

En los últimos años, las redes de telecomunicaciones han experimentado un crecimiento exponencial en el tráfico de datos debido, en gran medida, a la expansión de Internet.

1.1 Motivación

En actualidad los operadores están migrando hacia un esquema de red multicapa, donde se tiene equipamiento IP/MPLS para realizar la conmutación de paquetes mientras que se utiliza una red óptica como red de transporte que realiza una conmutación de circuitos [15].

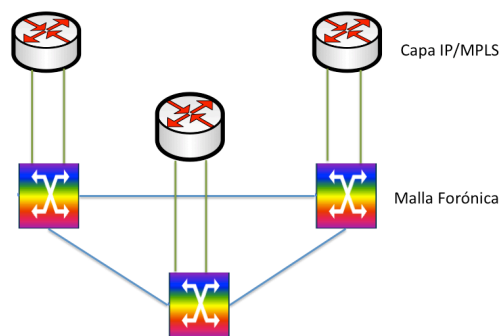


Figura 1-1: Red de Transporte de próxima generación

Este tipo de redes hacen uso de la multiplexación estadística a nivel IP/MPLS gracias a la conmutación de paquetes, mientras que se utiliza la capa óptica que es el único medio de transmisión capaz de enviar información hasta los terabits por segundo. Sin embargo, una vez está desplegada la planta de fibra, el operador debe sacarle el mayor rendimiento. Para ello, tareas como el cómputo de caminos se hace sumamente importante ya que permiten aprovechar de una forma óptima la planta de fibra [30].

Estudios previos [15][17][18], demuestran que el cálculo combinando la información de la capa IP y la capa óptica hacen que se reduzca el coste de la red y que se utilice de una forma más óptima.

En las redes de transporte, el desarrollo de arquitecturas en el plano de control [1][2][4] nos lleva a un escenario donde el cálculo de rutas se realiza de manera automática por la red, basándonos en el cálculo de ruta de forma distribuida mediante OSPF o en la arquitectura *Path Computation Element* (PCE) [13]. Esta arquitectura PCE se basa en un elemento con gran capacidad de computación para calcular LSPs (*Label Switched Paths*) o rutas de una red basándose en su grafo y la *Traffic Engineering Database* (TED) de dicha red, así como teniendo en cuenta el ancho de banda asignado u otros parámetros de *Quality of Service* (QoS). Esta arquitectura permite calcular rutas tanto en entornos de redes ópticas con conmutación de longitud de onda como en redes multicapa [13].

1.2 Objetivos

La finalidad de este Proyecto de fin de carrera es la implementación y evaluación de algoritmos en los PCEs de *Telefónica I+D* (TID) y del *Centre Tecnològic de Telecomunicacions de Catalunya* (CTTC). Los objetivos que se pretende ir alcanzando a lo largo de este proyecto son los siguientes:

1. Estudiar y analizar el estado del arte de los algoritmos de encaminamiento y de asignación de longitud de onda en redes de fibra óptica para un modelo de capa única.
2. Realizar una comparativa de algoritmos del estado del arte de cálculo de rutas y asignación de longitud de onda.
3. Estudiar y analizar el estado del arte de los algoritmos de encaminamiento *multicapa*.
4. Evaluar algoritmos multicapa para comprobar su rendimiento en escenarios de red realistas.

1.3 Estructura de esta memoria

Una vez obtenidos los objetivos planteados previamente, procedemos a la escritura de la memoria de este Proyecto Final de Carrera (PFC). Esta memoria queda estructurada de la manera siguiente:

1. El primer capítulo es una introducción al problema estudiado. Se exponen la motivación y los objetivos del proyecto, y se listan los medios con los que se dispone para la elaboración.
2. En el segundo apartado se estudia el estado del arte de las redes de próxima generación. Se introduce el modelo de red de próxima generación y su arquitectura, centrándonos posteriormente en la arquitectura GMPLS aplicada al plano de control. Posteriormente se explican los modelos de plano de control multicapa y finalmente explicamos la arquitectura de red basada en PCE para entornos multicapa. Finalmente se presenta y explica el funcionamiento del mecanismo VNTM.
3. En el tercer capítulo se presentan los dos escenarios utilizados para las simulaciones de este proyecto: el *testbed* del CTTC y el de TID. Para ambos casos se explican las características funcionales y las descripciones de plano de control. Además, se presentan las topologías utilizadas en cada caso con sus características.
4. En el apartado 4 se presentan las soluciones desarrolladas para el caso de redes de conmutación de longitud de onda (WSON). Se explican los diversos algoritmos utilizados y se muestran los resultados en términos de probabilidad de bloqueo. Se analizan los resultados y se realiza una comparación en términos de probabilidad de bloqueo.
5. En el apartado 5 se presentan las soluciones desarrolladas para el caso de redes multicapa (IP/MPLS sobre redes WSON). Se presentan los dos modelos de PCE a tener en cuenta: PCE multicapa con TED multicapa integrada y PCE multicapa con 2 TEDs (una por cada capa). Se explican las políticas de encaminamiento utilizadas en cada caso y se obtienen los resultados en términos de cantidad de conexiones y

ancho de banda establecidos en la red. Se obtienen también resultados de tiempos de computación de camino en cada caso. Finalmente se realiza una comparativa de todas las políticas y modelos utilizados.

6. El sexto capítulo finaliza este proyecto explicando las conclusiones obtenidas a partir de los resultados y propone futuras líneas de trabajo.
7. Por último, se encuentran los anexos que complementan todo este trabajo realizado. En el anexo A se presenta una validación de los resultados obtenidos, mostrando ejemplos de estabilidad de los valores numéricos resultantes de las simulaciones. El apartado B se trata de un manual de usuario, donde se explican los pasos necesarios para utilizar las implementaciones realizadas. Por último el anexo C muestra una publicación resultante del trabajo realizado en este PFC.

1.4 Medios a utilizar

Las herramientas necesarias para la elaboración del proyecto, disponibles en *TID* para su uso, son las siguientes:

- PC para el desarrollo y documentación del proyecto.
- Testbed del Centre *Tecnològic de Telecomunicacions* de Catalunya (CTTC).
- Testbed de Telefónica I+D (TID).
- Sistema operativo GNU/Linux.
- Entorno de desarrollo *Matlab* para la evaluación y muestra de resultados obtenidos.
- Procesador de textos.

2 Modelo de Red de nueva generación

A finales de los años 90, con el creciente aumento de redes WDM [35], los proveedores de servicios de red empiezan a pensar en la elaboración de un plano de control inteligente que sea capaz de simplificar el aprovisionamiento de la red, reducir el coste operacional y ofrecer la posibilidad de proveer nuevos servicios de red. La rápida evolución de las tecnologías ópticas hace posible que el ancho de banda que ofrece WDM sea aprovechado eliminando la necesidad del envío de paquetes por salto, y ha obligado a que todas las redes troncales sean ópticas.

Una red óptica [35] se compone de un conjunto de OXC (*Optical Cross Connects*) unidos formando una topología arbitraria, cuya función es la de proveer conectividad a diferentes subredes IP/MPLS. El mecanismo de transporte en estas redes, es una longitud de onda, es decir, una señal óptica. Con GMPLS, el *backbone* formado por los OXC y las subredes IP/MPLS comparte funciones comunes en el plano de control, permitiendo la integración de las redes ópticas en la estructura global de Internet.

El modelo de red ASON [3] (Automatically Switched Optical Network) es el resultado de este auge de las tecnologías de redes ópticas, y lo que se busca con esta arquitectura es el desarrollo de una red óptica conmutada en todos sus ámbitos: plano de control, datos y gestión. Esta arquitectura nos define los componentes de la capa óptica de control y sus interacciones, de tal forma que se permita el establecimiento y finalización de conexiones a partir de peticiones directas desde el usuario (cliente). Para soportar las posibles peticiones de diferentes clientes, la arquitectura define los componentes, puntos de referencia y reglas que hay que aplicar en las distintas interfaces entre los clientes y las redes (dominios) y entre distintas redes entre sí.

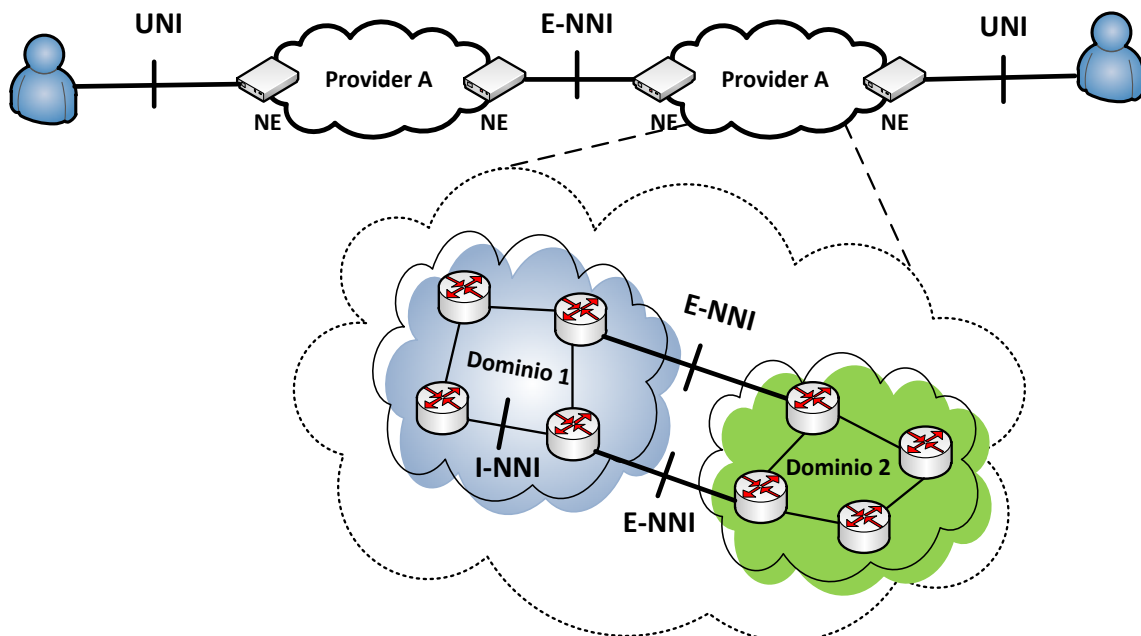


Figura 2-1: Esquema Modelo de red ASON

A continuación se explica la utilidad de los distintos puntos de referencia de la arquitectura [3]:

- *User-to-Network Interface* (UNI): se encuentra en el borde de las redes (dominios) y se utiliza para pedir un servicio punto a punto de esa red.
- *External Network-to-Network Interface* (E-NNI): se encuentra situado entre subredes o dominios de red y se encarga de transmitir la petición de servicio entre esas regiones.
- *Internal Network-to-Network Interface* (I-NNI): se encuentra situado entre los elementos de red de una subred y se encarga de que se realice el servicio a través de dicha subred.

GMPLS nos aporta el conjunto de protocolos que se encargan de organizar el funcionamiento y distribución de conexiones entre las distintas subredes que forman Internet. Cada red puede tener distintas capacidades de conmutación, por lo que GMPLS se encarga de gestionar todas ellas.

2.1 Arquitectura de red de próxima generación

Las redes de transporte de nueva generación están compuestas por un plano de control, un plano de gestión y un plano de datos. El plano de datos se utiliza para llevar a cabo la transmisión de los paquetes de información. Este plano está compuesto por enlaces de tipo óptico (fibras ópticas) o IP/MPLS, por los que se envían los paquetes de datos de los usuarios. El plano de gestión se encarga de numerosas tareas, entre las que se incluye la evaluación de la seguridad de la red, gestión de reportes del estado de red, etc. El plano de control se encarga del encaminamiento, monitorizar el estado de los links de la red y establecer o quitar conexiones en la red. Al contrario que el plano de gestión, estas tareas las realiza de una forma distribuida.

Mientras que ASON define una arquitectura, el plano de control GMPLS implementa unas funcionalidades. Mezclando las funcionalidades de GMPLS [2][5][35] junto con la arquitectura de ASON se puede conseguir una solución de plano de control para las redes de transporte. Con ello podemos realizar un establecimiento y gestión automatizado de conexiones en toda la red.

2.2 Plano de control: GMPLS

GMPLS [35][6] nace con la idea de poder aplicar un único plano de control universal para cualquier topología de transporte (redes WDM, redes TDM, etc.). Esta arquitectura tiene por objetivo generalizar el protocolo MPLS de forma que se pueda realizar conmutación de cualquier tipo de recursos. Los protocolos en los que se basa fueron diseñados e implementados para aplicar mecanismos Ingeniería de Tráfico a redes MPLS. Estos mecanismos se utilizan para dar la posibilidad de establecer tráfico de datos en un camino pre-computado en la red, de forma que podamos maximizar la utilización de los recursos de red disponibles. Así podemos encaminar tráfico evitando puntos saturados de la red o elegir enlaces que cumplan los requisitos de calidad de servicio.

En Figura 2-2 podemos ver un ejemplo de modelo de red GMPLS para una red de transporte óptica [40][6].

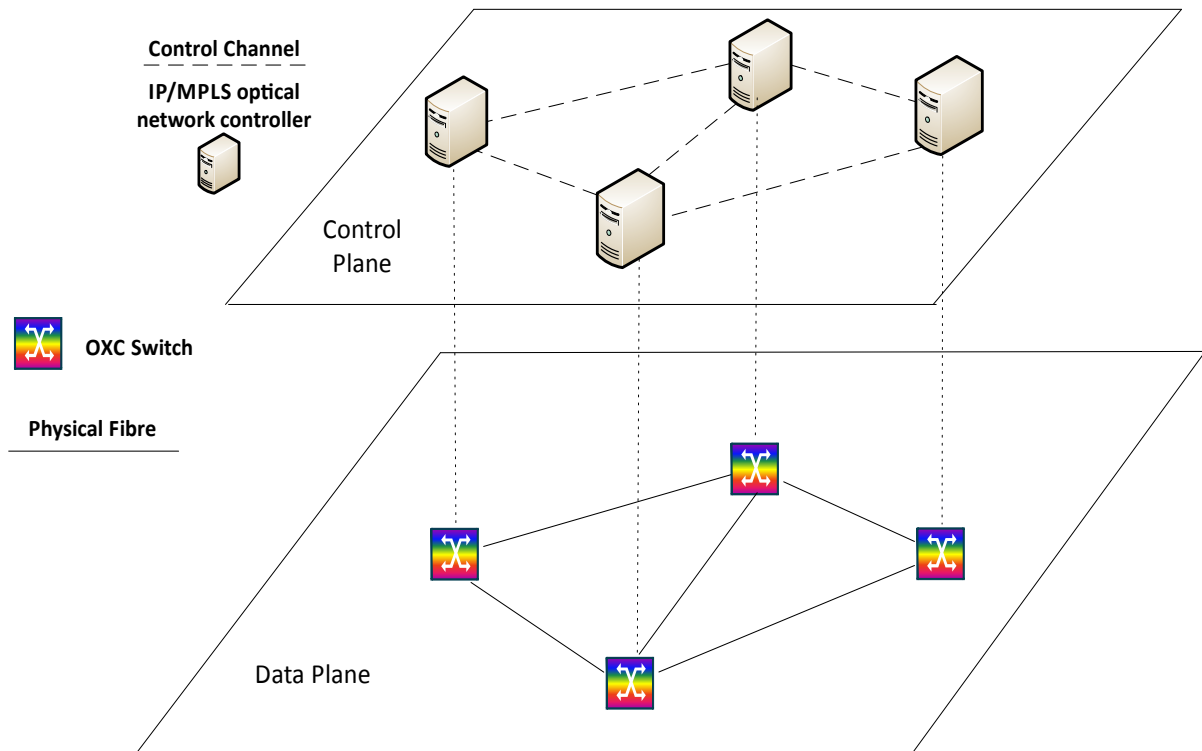


Figura 2-2: Modelo GMPLS basado en red óptica

GMPLS [1][2][5][35] extiende las funciones de MPLS para que se puedan soportar múltiples tipos de jerarquías de conmutación [2][34]. La Figura 2-3 muestra los distintos niveles de conmutación en GMPLS, estos son los siguientes junto con los tipos de routers Label Switch Routers (LSRs) que los soportan [35]:

- Packet Switch Capable (PSC): MPLS Routers.
- Layer-2 Switch Capable (L2SC): LSRs que transmiten datos basados en la VPI / VCI de ATM.
- Lambda switch Capable (LSC): Optical Cross-Connects que pueden operar al nivel de una longitud de onda individual.
- Time Division Multiplex Capable (TDM): Add-Drop Multiplexer (ADM).
- Fiber-Switch Capable (FSC): PXC o OXC que pueden operar al nivel de una o múltiples fibras.

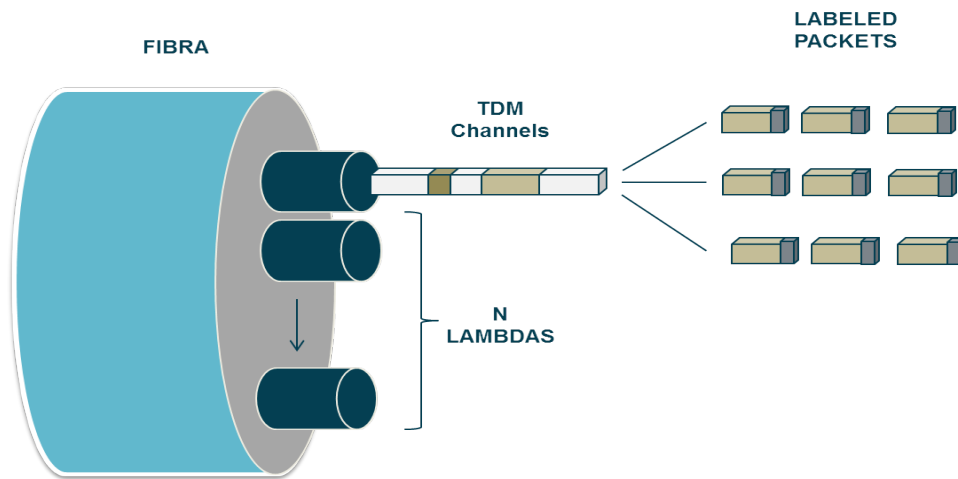


Figura 2-3: Jerarquía de tipos de conmutación

La jerarquía de GMPLS permite a la red operar con distintos tipos de conmutación, los cuales nos definen las unidades de datos que cada dispositivo puede manejar y conmutar. Es decir, nos definen el nivel al que se puede demultiplexar las señales de datos entrantes al nodo por una interfaz, conmutarlas y enviarlas por otra interfaz. Para crear dicha jerarquía se definen los siguientes procedimientos [34][35]:

- Un LSR crea un Label Switched Path (LSP) usando *Traffic Engineering Label Switched Path (TE LSP)*. Un Label Switch Router (LSR) es la manera de denominar a los routers que soportan GMPLS. Podemos ver los tipos que tenemos en la página anterior y también una ilustración de los mismos en la Figura 2-4.

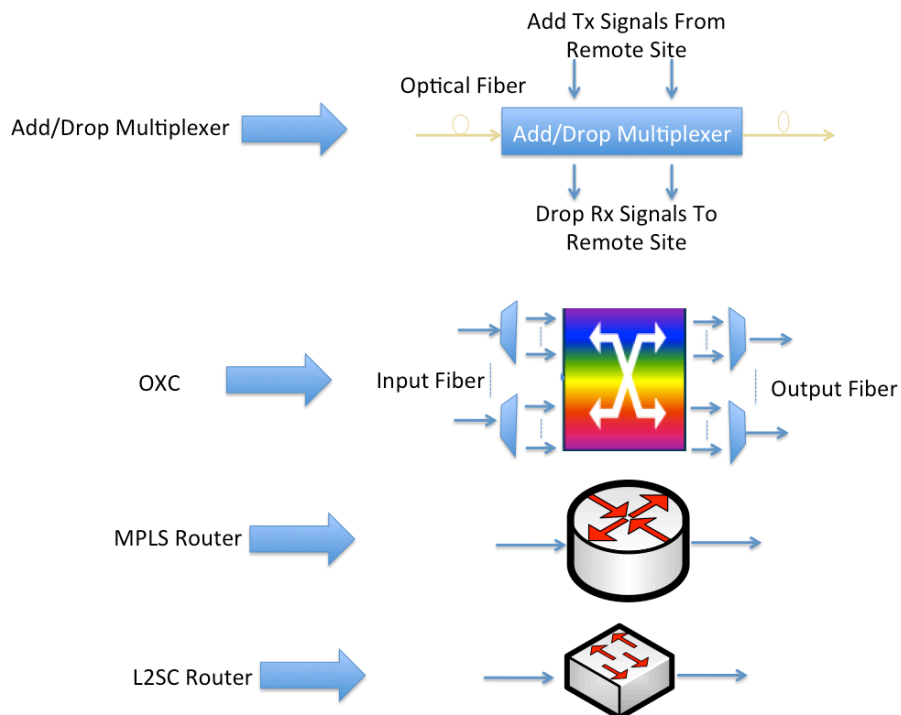


Figura 2-4: Tipos de routers (LSR)

- El LSR se encarga de formar una Forwarding Adjacency (FA) a partir de un LSP (notificando a este LSP como un TE Link en la misma instancia de OSPF (ver sección 2.2.1) que los links que se usaron para crear este LSP). Mencionar que esto solamente es posible en caso de que los LSPs que forman el FA están controlados por la misma instancia del plano de control, es decir, si el plano de control es integrado.
- Se permite a otros LSRs usar FAs para la computación de los caminos.
- Usando otros LSPs creados por otros LSRs para crear el suyo propio.

Para hacernos idea práctica, cada conexión que establecemos esta caracterizada por una tecnología de conmutación. En una red multicapa estas conexiones se pueden combinar para dar lugar a nuevas conexiones de diferentes tecnologías. No todas las combinaciones son posibles, por ello GMPLS define una jerarquía a partir de las tecnologías de plano de datos disponibles, y a partir de los interfaces de adaptación que hay definidos.

GMPLS tiene la función de cubrir las siguientes tareas [35]:

- Autodescubrimiento de la topología de la red. Cada vez que se conecta un nuevo elemento de red, se le notifica automáticamente información sobre sus vecinos.
- Anuncio de recursos disponibles. Además de tener información sobre la topología de la red, puede conocer el estado de los recursos disponibles en ésta.
- Gestión de los enlaces disponibles. Se encarga de la gestión de cada uno de los enlaces que forman parte de una ruta o LSP, de forma que tienen que ser establecidos, reservados o liberados según los requerimientos de la red.
- Encaminamiento de paquetes. Con la información que se posee del estado de la red, se realiza una computación de caminos mediante algoritmos para selección la ruta óptima, aplicando restricciones de ingeniería de tráfico.
- Gestión del camino. Incluye distribución de etiquetas, así como establecimiento, gestión y terminación de la ruta.

Estas funcionalidades están cubiertas por el plano de control [16], el cual se organiza con una serie de módulos que se muestran en el diagrama de la Figura 2-5:

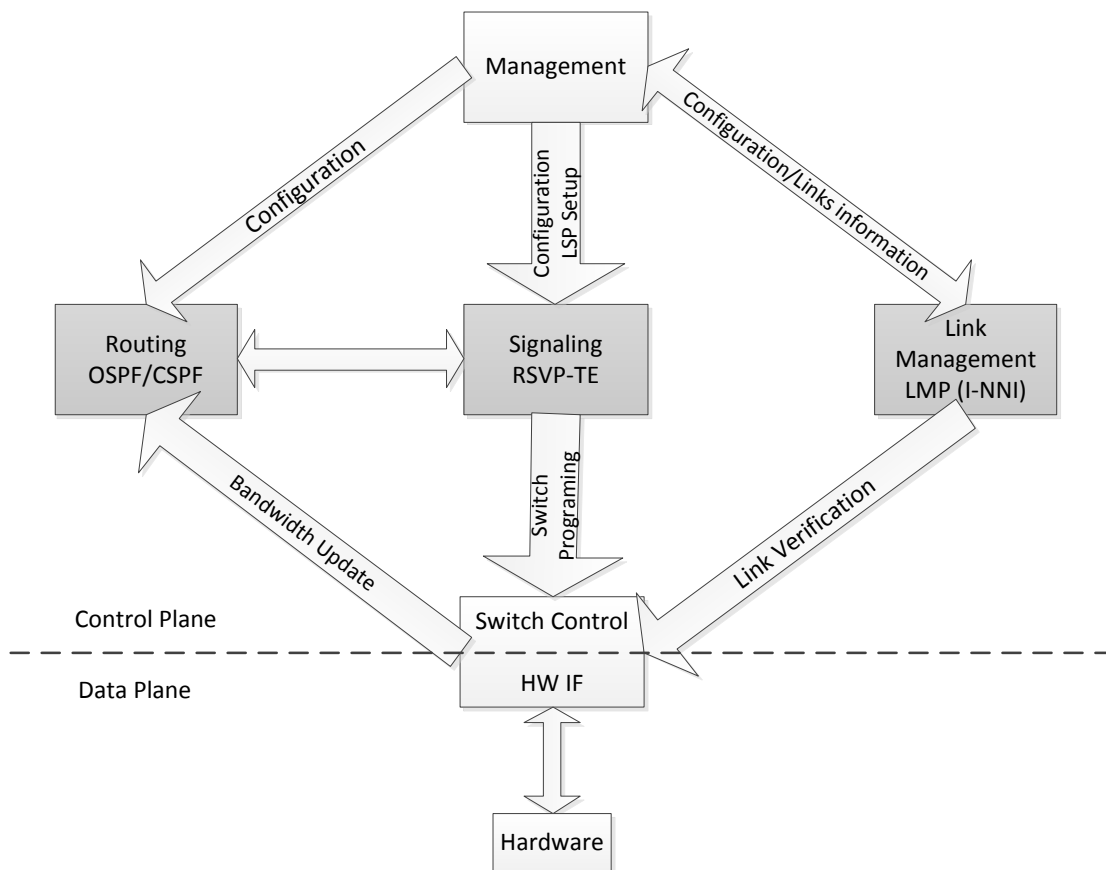


Figura 2-5: Diagrama de funcionalidad del plano de control [16]

GMPLS está compuesto por un conjunto de protocolos que se encargan de cumplimentar las funciones que se han descrito previamente. Entre estos protocolos son los siguientes, los cuales podemos observar también en la figura anterior con fondo gris [34][35]:

- RSVP-TE: se utiliza para la señalización de rutas mediante la reserva de recursos de la topología de red.
- OSPF: sirve para diseminar la información de la topología de red y su ingeniería de tráfico.
- LMP: se encarga de gestionar el estado de los enlaces de la red.

2.2.1 OSPF

El protocolo OSPF (*Open Short Path First*) se utiliza para diseminar la información de topología e ingeniería de tráfico (características y estado de los enlaces en cuanto a su uso) y construir una base de datos de ingeniería de tráfico [36]. OSPF es capaz de diseminar la información de la topología de red y la de la ingeniería de tráfico (por ejemplo ancho de banda disponible en los enlaces, grupos de riesgo compartido (SRLG) a los que pertenece, etc.). Mediante la información que se obtiene a partir de OSPF-TE se puede construir una base de datos de ingeniería de tráfico (*Traffic Engineering Database*). Para que OSPF pueda soportar GMPLS se definen una serie de mejoras en las propiedades de la Ingeniería de Tráfico de los GMPLS TE Links para que puedan ser notificados en OSPF-TE como

LSAs [37]. Mencionar que también se puede utilizar el protocolo ISIS-TE con la misma funcionalidad que OSPF-TE, es totalmente opcional el uso de uno u otro.

2.2.2 RSVP

El protocolo RSVP (*Resource reSerVation Protocol*) [38] se encarga de la señalización de rutas para la reserva de recursos de flujos de datos tanto *multicast* como *unicast*. A este protocolo se le han hecho extensiones de ingeniería de tráfico (TE) para que pueda soportar el establecimiento de túneles (LSPs) en MPLS [39]. Mediante esta extensión, RSVP-TE consigue la señalización de túneles LSP para que sean automática y fácilmente reconfigurables ante posibles incidentes en la red como fallos, congestión de red o cuellos de botella. También se han definido las extensiones necesarias de RSVP-TE para GMPLS [43]. Con estas extensiones, en GMPLS conseguimos definir los LSPs como TE-Links (enlaces entre dos nodos con propiedades de ingeniería de tráfico). Definiendo los LSPs de esta manera éstos pueden ser usados para formar otros LSPs, a los que se denomina en terminología MPLS/GMPLS “*Forwarding Adjacency (FA) Links*”. Cabe mencionar que esto solamente es posible si el plano de control es el mismo para los LSPs que forma el FA Link.

2.2.3 LMP

El protocolo LMP (*Link Management Protocol*) [26] es el protocolo de gestión de enlaces. Se encarga de gestionar el correcto funcionamiento de los enlaces, verificación de los enlaces y comprueba conectividad entre nodos adyacentes. Para el establecimiento de una sesión LMP tenemos que tener siempre disponible un canal físico para utilizarlo como canal de control. A continuación se detalla el funcionamiento de este protocolo para la verificación de conectividad de un TE Link [27].

Queremos añadir un nuevo link entre dos nodos y queremos verificar su conectividad. Este link se basa en 3 *data links* que utilizan tres puertos en cada nodo (ver Figura 2-6):

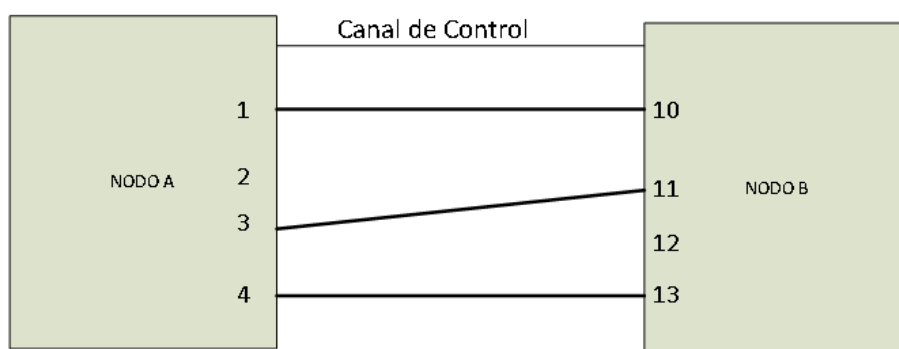


Figura 2-6: Link entre los nodos A y B

A continuación mostramos la secuencia de mensajes que se intercambian entre los dos nodos en la Figura 2-7. Lo primero que tienen que conocer son las interfaces usadas por los *data links* a ambos extremos del enlace para poder enviarse los *Test Messages*, por lo que se intercambian una serie de mensajes por un canal de control bidireccional para poder realizar el mapeo de interfaces o puertos en cada extremo de los data links (tendremos 2 *Interface_IDs* por cada data link que contenga el *TE_Link*).

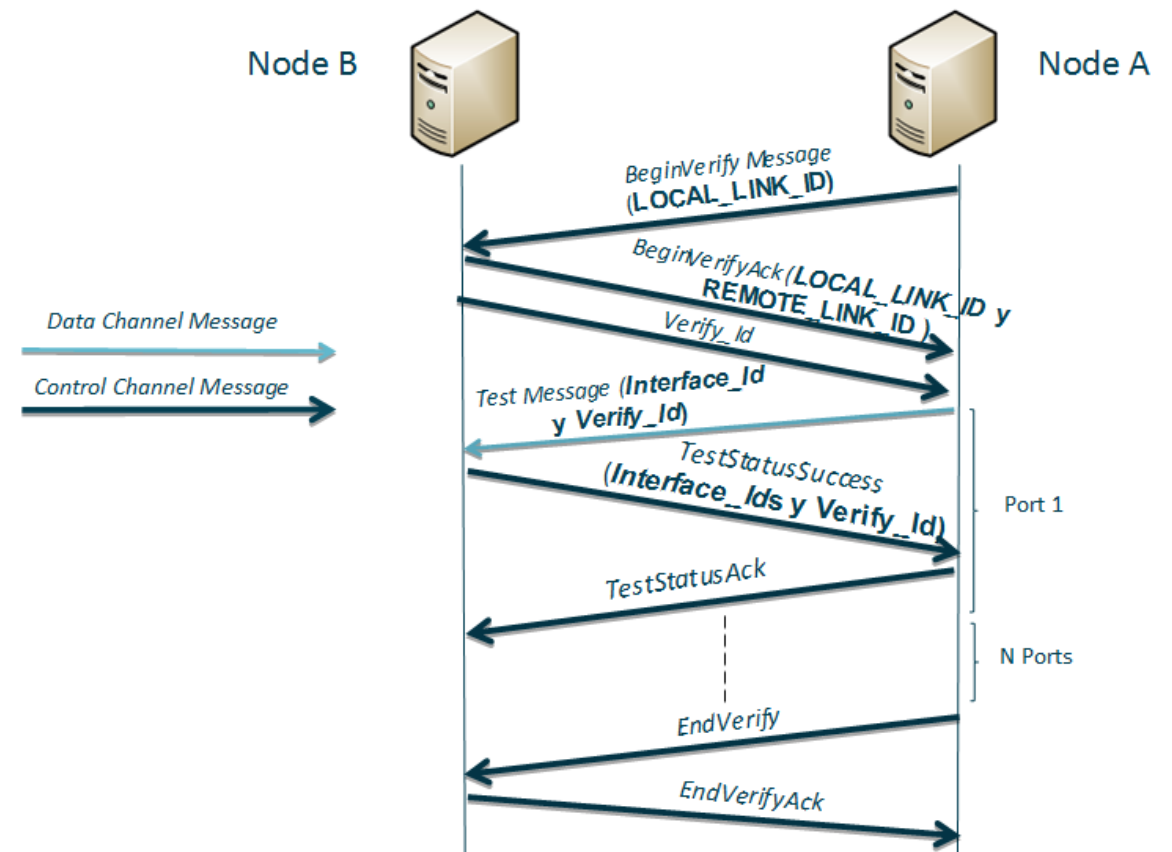


Figura 2-7: Intercambio de mensajes entre nodos para comprobar la conectividad del link

La figura 2-7 hace ilustración del siguiente proceso para realizar el establecimiento y verificación del nuevo link:

1. El nodo A envía un mensaje *BeginVerify* al nodo B a través del canal de control. En este mensaje se incluye el identificador local (dirección IP o interfaz) que A asigna al *Link*.
2. Cuando B recibe el mensaje crea un objeto *Verify_Id* y lo asocia al *TE_Link* de A. Esta asignación nos sirve luego cuando B recibe los *Test Messages* de A, los cuales incluyen el objeto anteriormente mencionado. De esta manera B descubre también el identificador que A ha asignado al *TE_Link*. Posteriormente B contesta con un *BeginVerifyAck* que lleva el identificador que B asigna al *TE_Link*. También se incluye un objeto *REMOTE_LINK_ID* con la asociación de *Link_Ids* asignada por ambos. El objeto *Verify_Id* también es incluido en este mensaje, el cual se envía a través del canal de control.
3. Cuando A recibe el mensaje previamente enviado, empieza a enviar *Test messages* periódicos que incluyen el objeto *Interface_Id* para el puerto usado y el objeto *Verify_Id* que fue creado por B.
4. Cuando B recibe el *Test message* mapea el *Interface_Id* enviado (*Interface_Id* = 1) con el suyo propio (*Interface_Id* = 10) y envía de vuelta un mensaje *TestStatusSuccess*, el cuál incluye ambas *Interface_Ids* para los puertos utilizados junto con el objeto *Verify_Id*. Con este objeto el nodo A puede determinar los identificadores del *TE_Link* a los que pertenece el data link.

5. El nodo A devolverá un mensaje *TestStatusAck* por el canal de control indicando que ha recibido el *TestStatusSuccess*.
6. Este proceso se repite hasta que todos los puertos de los data Links han sido comprobados.
7. Una vez comprobados el nodo A envía un mensaje *EndVerify* por el canal de control indicando que la comprobación ha terminado.
8. El nodo B responderá con un mensaje *EndVerifyAck*.

2.3 Plano de Control Multicapa

El mantenimiento y control de las redes de telecomunicación implica la interacción humana para crear, establecer y mantener nuevos servicios. De esta manera el plano de control aparece como un conjunto de funciones (encaminamiento, reserva de recursos y gestión de los links de la red) [24], las cuales se realizan de una manera automática y distribuida en la red. Este proceso automático permite al operador de red reducir la complejidad de la gestión de la misma. Actualmente en las redes multicapa, el plano de control opera de manera separada en cada capa, sin embargo existen distintos tipos de plano de control dependiendo del nivel de comunicación entre las capas de la red. Más adelante se explicará el funcionamiento de estos distintos modelos y los requisitos necesarios para el correcto funcionamiento de cada uno.

Las funcionalidades del plano de control se consiguen mediante la utilización de la arquitectura GMPLS, la cuál también permite la utilización de la jerarquía de tipos de conmutación (previamente explicada en la sección 2.1). Para que el plano de control pueda operar correctamente en entornos multicapa debe ser capaz de resolver las siguientes tareas:

- Relacionar e informar de las conexiones entre las interfaces de datos de los elementos de la red. Estas conexiones pueden ser dentro de una misma capa o entre capas.
- Proporcionar una interfaz para solicitar caminos entre ambas capas.
- Proporcionar un método que permita señalar e informar de las reservas de recursos y cambios en las mismas.

Como podemos ver el plano de control para redes multicapa presenta importantes retos con respecto a los planos de control con una única capa. Para que las tareas mencionadas sean factibles, hay dos clases de aproximaciones principales, un modelo “*overlay*” con separación de información sobre cada capa, y el otro modelo donde tenemos mayor relación de confianza entre capas y mayor aprovechamiento de recursos a nivel global de toda la red (modelo *Peer*).

En el modelo “*overlay*” la relación de confianza entre ambas capas está limitada, y por tanto, el intercambio de información en el plano de control entre equipos de ambas redes es reducido. De esta forma se tienen que emplear los protocolos con las opciones totalmente estándar, habitualmente en una versión limitada. La otra aproximación asume una relación de confianza mayor, tanto entre ambas capas, como entre los fabricantes de los nodos de ambas capas, de tal forma que se puede emplear los protocolos con todas las extensiones necesarias y toda la información que se desee para así lograr un mejor comportamiento y aprovechamiento de los recursos. Esto facilita el uso de los protocolos de GMPLS con todas sus extensiones, así como de toda la información deseada para alcanzar la utilización

óptima de los recursos en el plano de control. Para hacer esto posible es necesario mejorar la comunicación entre capas y todos los elementos de la red.

Como hemos mencionado, al juntar capas con modelos de control relativamente distintos surgen varios retos. Un ejemplo claro es el paso de un plano de control en banda (MPLS) a uno fuera de banda (GMPLS). En una red IP/MPLS las interfaces de datos son utilizadas para emitir y recibir los mensajes de plano de control. Esto supone que en el momento en que dos interfaces de dos equipos se interconectan (y son compatibles) el plano de control anuncia esta conexión como por ejemplo en OSPF. Simplemente el hecho de conectar dos equipos a través dos interfaces hace que ambos conozcan su existencia y los destinos que se pueden alcanzar a través de dichas interfaces. Esto supone una facilidad y comodidad desde el punto de vista de uso y configuración pero no es viable en entornos donde existe un plano de control fuera de banda. En estos casos, hay que incluir mecanismos adicionales que describan y mantengan dichos enlaces.

Por ejemplo, una red de transporte necesita configuración adicional, como las longitudes de onda que se van a utilizar para transmitir información entre dos interfaces. En este caso, el proceso de diseminación de la información y configuración cambia sustancialmente. Para que esto pueda realizarse con el plano de control, es necesaria una señalización fuera de banda, lo que quiere decir que se necesita una red alternativa establecida previamente para intercambiar los mensajes de plano de control ya que las interfaces del plano de datos no podrán realizar este papel por lo ya comentado anteriormente. Esta señalización fuera de banda puede consistir en una red Ethernet que interconecte los equipos, una longitud de onda configurada por gestión para transmitir el plano de control... Es necesario el establecimiento de una red que permita la información de adyacencias, las solicitudes de reserva de recursos y demás operaciones realizables mediante plano de control.

En las redes multicapa, siguiendo el modelo “*overlay*”, la relación entre capas de red y de transporte no puede ser establecida automáticamente al conectar las interfaces de equipos de ambas capas por las particularidades de cada capa. En este caso, el plano de control necesita tener una conexión directa entre todas las capas, de tal forma que pueda operar, establecer y reservar recursos en la red. Para esta interconexión física de capa se utilizan TE-Links *Inter-layer*.

Para estas conexiones del plano de control que acabamos de mencionar, necesitamos establecer canales en ambas direcciones, de tal forma que mediante el protocolo LMP podemos informar de las tecnologías de conmutación permitidas por los terminales de las capas interconectadas. De esta manera es totalmente posible establecer automáticamente un TE Link *inter-layer* con la información del plano de control mediante la información del protocolo LMP.

Para comparar ambos modelos de plano de control en redes multicapa de una manera práctica, se explican ambos casos para crear caminos con PCE. De manera que también podemos estudiar ambos modelos en términos de rendimiento y optimización de recursos para la red:

- En caso de que la topología de red óptica no sea visible desde la capa IP/MPLS, los LSPs en la capa óptica se notifican como *virtual TE-links* en la capa IP/MPLS. La topología TE de la capa IP/MPLS es una topología virtual y como la topología de red óptica no es visible no se pueden crear caminos *end-to-end* óptimos. El

problema de esto es que se pueden malgastar los recursos disponibles en la capa óptica. Para este caso necesitaríamos disponer de un PCE centralizado para cada capa y establecer un modelo de coordinación entre ellos.

- En caso de que la información de TE de la topología óptica sí sea visible desde la capa IP/MPLS, sí que se pueden computar caminos *end-to-end* óptimos teniendo en cuenta ambas topologías red. Los links deben ser notificados como links con múltiples capacidades de conmutación en el plano de control. En este caso utilizaríamos un PCE único distribuido que tenga información global de la red para todas las capas de ésta. Como ya veremos más adelante, este modelo se ha establecido como el más óptimo y el que mayor escalabilidad aporta a la arquitectura.

Cabe mencionar que un “Virtual TE-Link” [9] se define como un TE-Link en la red de la capa del cliente. Este TE-Link se basa en recursos disponibles en la capa de transporte, que tienen que estar necesariamente reservados para dar soporte a este Link. Las dos principales propiedades de un “Virtual TE Link” son [22]:

- Se identifica como un TE-Link real, y por lo tanto contribuye a la construcción de la topología de red de la capa del cliente.
- No requiere la utilización de recursos en la capa de transporte hasta que se utiliza, por lo tanto puede compartir recursos de esa capa con otros “Virtual TE-Links”.

2.3.1 Modelos de plano de control multicapa

Como hemos mencionado en el apartado anterior, para redes IP sobre WDM tenemos 3 modelos de plano de control multicapa distintos [24][30]. Estos modelos son: *overlay*, *peer model* y *border peer model*.

2.3.1.1 Modelo Overlay

En el modelo *overlay* [25][30] nos encontramos con un modelo de capas separadas (cliente-servidor). La parte de red IP/MPLS es considerada como el dominio del cliente y la parte de red óptica de transporte se considera el dominio del servidor. Mediante los TE-Links Multicapa (*Inter-layer*), podemos establecer una conexión física entre capas, pero tenemos que ver cómo podemos realizar la reserva de recursos en ambas capas. La mayor característica de las redes multicapa es poder crear links y adyacencias entre terminales de la capa superior, basándonos en conexiones y tecnologías establecidas en capas inferiores. Esto quiere decir que la reserva de recursos ha de hacerse en diferentes capas, de tal forma que podamos tener un cierto ancho de banda y un mínimo *QoS* a la conexión de red establecida en la capa superior.

Como la relación entre la capa de transporte y la capa cliente no puede ser establecida automáticamente, necesitamos hacer uso de una interfaz capaz de realizar peticiones de conexión de una capa a otra, y encargarse de facilitar la reserva de recursos en ambas capas. Podemos conseguir esto a través del uso del modelo UNI (*User to Network Interface*) [3][28], el cual es una interfaz estandarizada por el IETF para establecer protocolos y el modelo de trabajo para servir estas funcionalidades descritas. El modelo UNI se divide en dos partes [30]:

- La parte del cliente UNI-C.
- Parte del servidor (*network*) UNI-N.

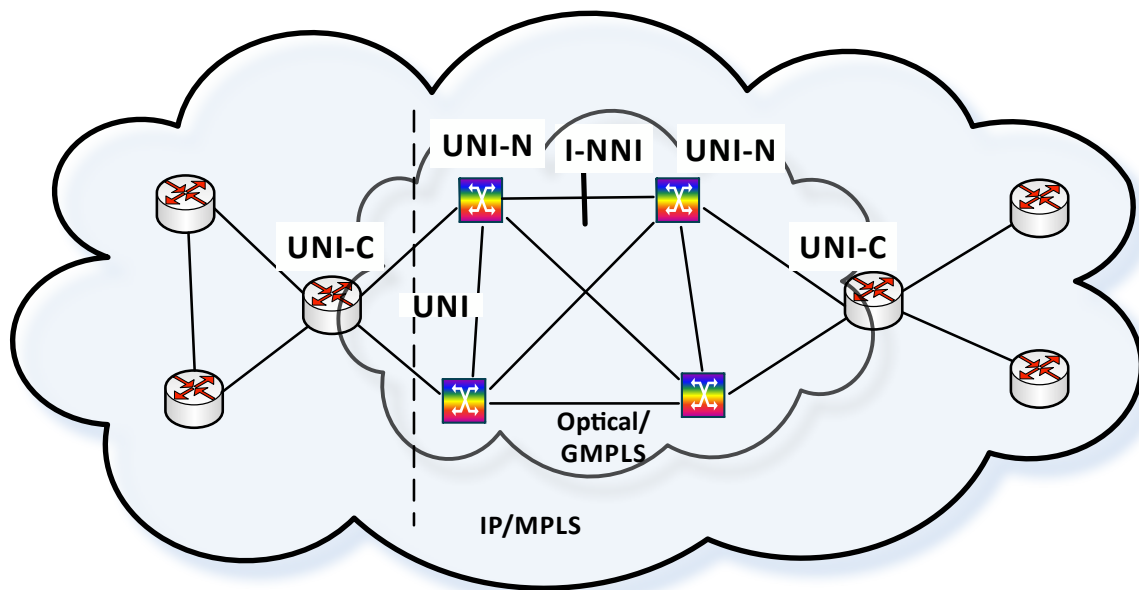


Figura 2-8: Arquitectura modelo *Overlay* [30]

UNI trabaja con RSVP-TE (y las extensiones para GMPLS) para realizar la reserva de recursos [29], y con OSPF-TE para notificar las nuevas adyacencias en la capa cliente después de realizar la reserva de recursos en la capa de transporte.

Cuando se solicita un nuevo LSP al plano de control en la capa de transporte, es necesario incluir los siguientes parámetros en la respuesta del PCE (PCRep) [9]:

- *La capacidad de conmutación del LSP.* Cuando creamos una nueva conexión punto a punto entre dos *routers*, esta conexión necesita usar interfaces en ambos dispositivos. Cada interfaz se caracteriza por un tipo de capacidad de conmutación, lo que significa distintas tecnologías de conmutación, las cuales caracterizan a la conexión establecida.
- Debe contener al menos un objeto RP (*Request Parameters*). Dentro de este objeto se encuentra el campo de los *end-points* y el objeto Request-ID-Number, que contiene el identificador dado para la petición.
- Si el cómputo de camino se obtiene correctamente devolvemos el objeto PATH. Dentro del mismo tenemos el camino a reservar dado por el objeto *ERO* (*Explicit Route Object*). Este objeto define los saltos en el camino y es opcional. En el modelo actual de capas separadas, el router que realiza la petición de camino no tiene toda la información de la red, de tal forma que no puede especificar el camino a establecer en la capa de transporte. El ERO en el caso multicapa necesita incluir los identificadores de los TE-Links del router que realiza la petición, el identificador del terminal conectado en la capa inferior y el identificador del terminal de destino (ambos terminales, el situado en la capa de transporte y el situado en la capa IP/MPLS).
- Dentro del objeto PATH devolvemos también el objeto BANDWIDTH, en caso de que en la petición hubiese ancho de banda requerido.
- Si el cómputo del camino pedido no ha podido ser resuelto hay que incluir un objeto NO-PATH.

Una vez hemos establecido el LSP en la capa de transporte, utilizamos la jerarquía de LSPs para definir la reserva de recursos con mayor granularidad e independientemente de la capa inferior. La jerarquía ayuda a dar una visión lógica de todas las conexiones en las diferentes capas de la red. Además, este modelo aumenta la escalabilidad de la red.

Como estábamos explicando, necesitamos reservar una capacidad determinada para el *Packet Switching Capable (PSC)* LSP sobre el LSP de la capa de transporte, que se caracteriza por ser LSC (*Lambda Switching Capable*). Esto hace posible crear diferentes adyacencias sobre este enlace con diferentes características, como por ejemplo diferente ancho de banda, diferentes prioridades o diferentes factores para la operatividad en la red. Cuando ya hemos establecido el LSP (la adyacencia) en la capa IP/MPLS es posible hacer uso de esta conexión sin tener en cuenta la capa de transporte. Esta adyacencia se usa reservando el ancho de banda necesario para establecer la conexión.

A pesar de todo este modelo de plano de control para redes multicapa, en cooperación con el UNI, tiene sus limitaciones:

- No es posible alcanzar la utilización óptima de los recursos en toda la red.
- En caso de proveer un servicio de restauración, el modelo UNI no tiene ningún mecanismo para asegurar que la restauración del camino no comparta links con ningún camino original previamente establecido.

Resumiendo, usando un modelo de plano de control como el *Overlay* (no integrado) no es posible tener una visión completa de toda la red. Por lo tanto, el plano de control debe desarrollar las siguientes características para obtener la coordinación óptima entre capas:

- Mecanismos de señalización UNI para establecer LSPs en las capas inferiores.
- Un PCE externo que debe de ser capaz de comunicarse con el plano de control de cada capa.
- El UNI debe ser capaz de intercambiar información de tráfico extendida.

2.3.1.2 Modelo Peer: Modelo integrado

En el modelo *peer* [30], todos los nodos soportan funcionalidades GMPLS para controlar todas las capas que haya en la red. De esta manera la información de la topología de red puede ser diseminada y comunicada a todos los nodos a través del protocolo OSPF-TE en el plano de control unificado. Cada nodo ha de interpretar la información de la topología de red para computar caminos incluyendo redes con diferentes capacidades de conmutación (ver Figura 2-9). De modo que se podrán crear LSPs para crear servicios *end-to-end*. Las principales ventajas de este modelo son el mantenimiento de la red, la seguridad y la escalabilidad.

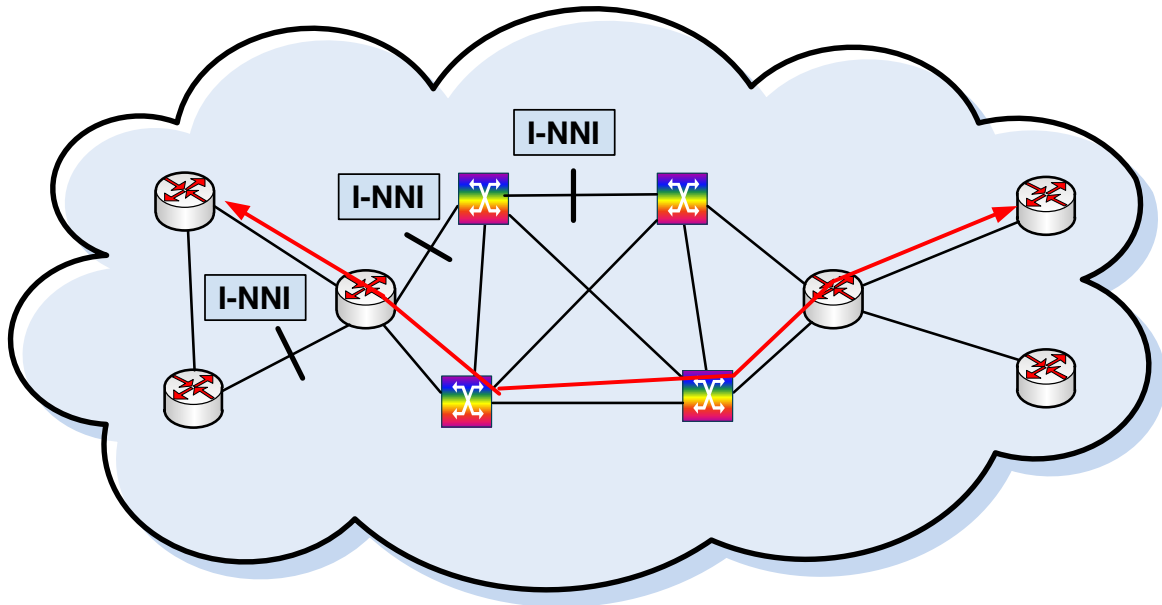


Figura 2-9: Esquema modelo Peer [30]

En caso de tener nodos integrados, el plano de control puede cambiar. En este caso dejamos de tener un modelo cliente/servidor, los terminales trabajan en ambos sentidos interpretando peticiones como servidor y enviándolas como cliente. Este tipo de modelo de trabajo figura en los estándares como la interfaz I-NNI (Internal-Network to Network Interface), en vez del modelo UNI. El uso del I-NNI se reduce a un dominio, de tal forma que tenemos un plano de control único e integrado en cada terminal. Este modelo de plano de control se conoce como *Peer Model* y todos los terminales en la red trabajan con la misma instancia del plano de control.

2.3.1.3 Border Peer Model

En el *Border peer model* [30], los routers con conexiones directas a nodos ópticos se denominan como *Border Routers (BR)* (Figura 2-10). En este contexto estos routers tienen dos funcionalidades: la primera como un router involucrado en el control sobre dominios GMPLS y la segunda como nodo óptico involucrado en el control sobre dominios GMPLS. Si un BR soporta GMPLS y MPLS, puede mantener información de Ingeniería de tráfico (*Traffic Engineering*) de la topología, tanto de la red IP como de la red óptica. El resto de routers y nodos ópticos guardan únicamente la información correspondiente a la capa a la que pertenecen.

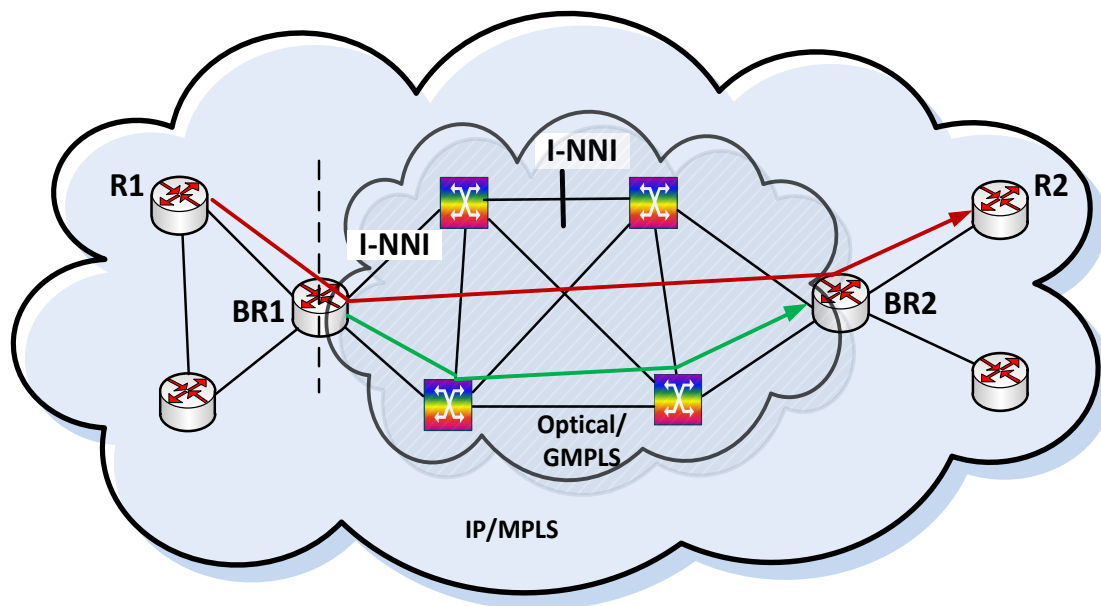


Figura 2-10: Modelo Border Peer [30]

Un BR puede computar caminos a través de la capa óptica incluyendo separación entre caminos de datos y de protección, basándose en la topología de red óptica. Si queremos crear una conexión *end-to-end* entre routers que no sean de borde, tenemos que determinar qué routers BR son más óptimos para crear el camino óptico entre ellos.

Para el despliegue de este modelo se pide que los routers y nodos ópticos sean suministrados por el mismo proveedor.

2.3.2 Requisitos y funcionalidades del plano de control integrado

Un plano de control integrado debe tener las siguientes funcionalidades:

- Funciones de provisión:
 - Debe permitir establecer de manera automática caminos en una capa, empleando los recursos existentes en dicha capa.
 - Debe de permitir la creación de H-LSPs de diferentes capacidades de conmutación (*Switching Capabilities*), capaces de albergar LSPs jerárquicos.
 - Se ha de poner incluir explícitamente una ruta por parte del operador (típicamente de planificación).
 - Se ha de poder crear, en una misma operación el LSP jerárquico.
 - El plano de control ha de interpretar los caminos multicapa obtenidos por planificación o un PCE.
 - El plano de control ha de poder crear el LSP tanto si se proporciona una ruta como si la ha de calcular en tiempo real.
- Encaminamiento y cálculo de camino
 - El plano de control ha de poder tener una visión topológica y de ingeniería de tráfico conjunta de ambas capas
 - El plano de control ha de poder construir dicha visión de manera autónoma. Para ello:

- Ha de anunciar unívocamente las asociaciones de interfaces entre capas (TE-Links *Inter-layer*)
 - Ha de ser capaz de recolectar el protocolo IGP de ambas capas
- Debe proporcionar mecanismos de cálculo de caminos óptimos multicapa (ML-PCE).
- Interoperabilidad
 - Debe permitir tanto la interoperabilidad entre equipos de distintos fabricantes en la misma capa.
 - Debe permitir la interoperabilidad entre fabricantes de diferentes capas.
 - Debe proporcionar mecanismos para controlar capas que posean PCE y capas que no lo posean.
 - Debe definir las interfaces de creación de las bases de datos multicapa ML-TED.

2.4 Path Computation Element

2.4.1 Arquitectura general

El *Path Computation Element* (PCE) [8][13] es una entidad capaz de computar un camino de red o una ruta basándose en el grafo de la topología de red y aplicando restricciones de computación durante el cálculo.

Son tres las tareas principales que debe realizar un PCE en una red óptica: calcular la ruta, asignar una longitud de onda, y realizar la validación física del camino. Una de las ventajas del PCE es su flexibilidad, ya que permite utilizar distintos algoritmos de encaminamiento, así como permitir la interconexión de redes en entornos multicapa y multi-dominio.

La arquitectura básica de un modelo basado en PCE está compuesta por tres elementos fundamentales: Path Computation Client (PCC), Path Computation Element (PCE) y Base de Datos de Ingeniería de Tráfico (Traffic Engineering Database - TED). En el ámbito de las redes MPLS y GMPLS, un PCC es un LSR (router MPLS/GMPLS) de la red que solicita el cálculo de un TE LSP a un PCE, y este se encarga de calcularlo y de proporcionárselo al PCC. Para el cálculo de caminos, un PCE utiliza la información de ingeniería de tráfico contenida en la TED. En este caso el PCE forma parte del plano de control, véase la Figura 2-11.

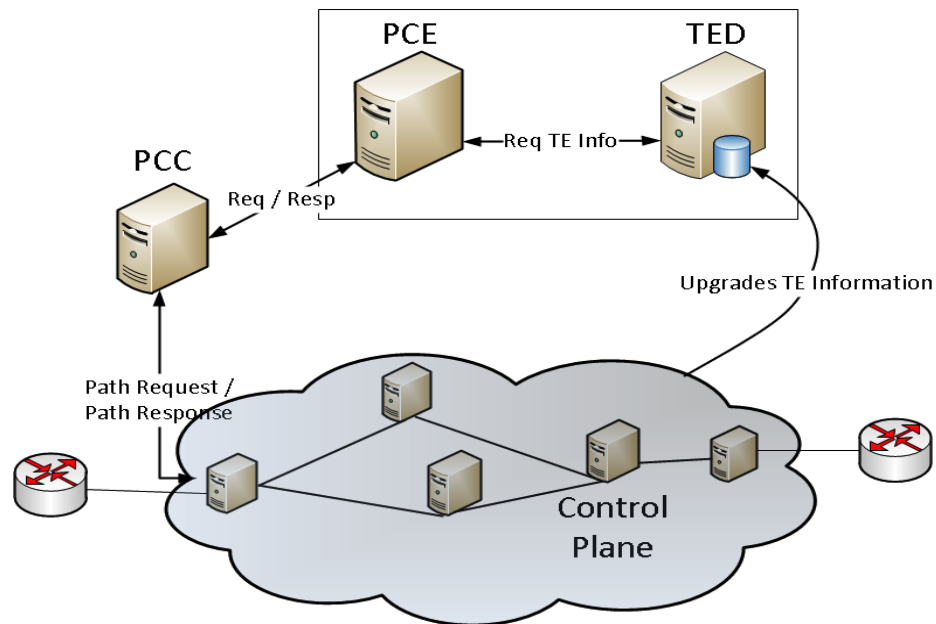


Figura 2-11: Diagrama de funcionalidad PCE

PCEP es un protocolo basado en el modelo de solicitud/respuesta y se utiliza para las comunicaciones entre un PCC y un PCE o entre dos PCEs [8][42]. Estas interacciones incluyen las peticiones de computación de camino, así como todas las notificaciones relativas al estado y configuración del PCE en el contexto de Ingeniería de tráfico aplicada a MPLS o GMPLS. El PCE que corresponde a un PCC puede ser manualmente adjudicado en la configuración o descubierto automáticamente [11].

Este protocolo diseñado para que permita la implementación de nuevos mensajes y objetos de forma que se puedan satisfacer nuevos requerimientos que puedan surgir en el futuro. Los mensajes PCEP actualmente definidos son los siguientes [42]:

- *Open y Keepalive Messages*: se utilizan para iniciar y mantener la sesión PCEP. Para inicializar una sesión necesitamos crear una conexión TCP y posteriormente establecer la conexión PCEP con ciertos parámetros sobre esa conexión creada. Los parámetros de la sesión se transmiten mediante los *Open Messages*. Los mensajes *Keepalive* se emplean para confirmar los *Open Messages*, y se utilizan una vez la sesión ha sido establecida para comprobar que la sesión sigue activa y disponible para su uso.
- *PCReq*: es el mensaje que se envía del PCC al PCE para solicitar un cómputo de camino.
- *PCRep*: es el mensaje que se envía del PCE al PCC en respuesta a la petición de cómputo de camino. Puede contener, como se especifica más adelante, un conjunto de rutas computadas (caso de encontrar camino) o una respuesta negativa en caso de que el PCE no encuentre camino alguno.
- *PCNtf*: mensaje que se envía del PCC al PCE o del PCE al PCC para notificar algún evento ocurrido.
- *PCErr*: mensaje que se envía en caso de ocurrir algún error en la comunicación.
- *Close message*: mensaje que se envía para cerrar la sesión PCEP. Si el PCC cierra la sesión el PCE borra todas las peticiones de cómputo de camino que tiene

pendientes. Si al contrario, es el PCE el que cierra la sesión, el PCC elimina todos los estados relacionados con solicitudes pendientes enviadas al PCE.

Los requerimientos básicos para este protocolo son los siguientes [10]:

- Se debe definir un protocolo individual para cada comunicación PCC-PCE o PCE-PCE.
- La comunicación se define como un modelo de cliente/servidor. Es decir, el protocolo debe permitir al PCC enviar peticiones de conmutación de caminos al PCE, y éste debe contestar con un mensaje de respuesta una vez el camino ha sido computado.
- El PCEP debe utilizar un protocolo de transporte ya existente que posea control de congestión. Para ello utilizamos conexiones TCP.
- El mensaje de petición de conmutación (*Request Message*) de camino ha de incluir las direcciones IP de los nodos origen y destino. Además debe permitir la inclusión de algunas restricciones de camino, tales como el ancho de banda necesario para la conexión o limitaciones de recursos (número de saltos, métrica, inclusión/exclusión de ciertos nodos/links, etc.). También el cliente tiene que tener la posibilidad de seleccionar el código OF (*Objective Function*) del algoritmo que el PCE ha de utilizar para la computación del camino. Además se tiene que poder asignar un nivel de prioridad a la petición para que, en caso de saturación de llegada de mensajes al PCE, se puedan seleccionar éstos según los niveles de prioridad adjudicados.
- El mensaje de respuesta del PCE (*Path Computation Response*) debe permitir al PCE incluir varios elementos, incluyendo al menos el camino computado. El camino computado ha de ser convertido al objeto *Explicit Route Object (ERO)* para el uso den la señalización del protocolo RSVP-TE. Una respuesta positiva del PCE debe permitir la inclusión de un *set* de propiedades sobre el camino computado, tales como: costes del camino (ej. métricas de los links) o ancho de banda computado en el camino. En caso de que el camino no pueda ser computado tenemos que enviar una respuesta negativa. Esta respuesta puede incluir detalles con las razones por las que no se ha encontrado un camino, por lo que se pueden modificar algunos parámetros para encontrar la posibilidad de encontrar el camino. También se permite la posibilidad de enviar varias peticiones de conmutación de camino agregadas en la misma petición, por lo que el PCE puede también contestar con varias computaciones de caminos disjuntos en el mismo mensaje de respuesta.
- El protocolo PCEP ha de contar con un mecanismo de detección y notificación de mensajes perdidos o corruptos. En caso de un mensaje no llegue correctamente, se debe activar un método de reenvío automático, para lo que también tenemos que disponer de un sistema de control de mensajes duplicados por si es necesaria esta retransmisión.
- La comunicación entre el PCC y el PCE es asíncrona. Es decir, el PCC no tiene que esperar a que le llegue una respuesta del PCE para poder enviar una nueva petición.

2.4.2 Modelos de funcionamiento del PCE en entornos multicapa

Uno de los problemas de tener la separación entre capas con dos planos de control independientes viene del cómputo de rutas multicapa [12][42]. Al no existir un plano de control integrado, los planos de control de cada capa sólo tienen visibilidad de los equipos de su misma capa proporcionando un encaminamiento en el mejor de los casos sub-óptimo.

Como hemos explicado previamente, el PCE es una entidad de la red dotada de un interfaz estándar de comunicación, el protocolo PCEP, la comunicación con los PCCs para recibir peticiones de conmutación de caminos. El PCE obtiene la información de ingeniería de tráfico de la red que controla creando la TED en la cual se recoge la información necesaria para calcular rutas dentro del dominio al que pertenece la entidad (recursos ocupados, libres, número de saltos...).

En la Figura 2-12 se presenta un modelo de red con PCEs en cada capa lo que permitiría el cálculo de rutas mediante un procedimiento estándar optimizando los recursos de cada capa por separado atendiendo a los criterios definidos por el operador [12].

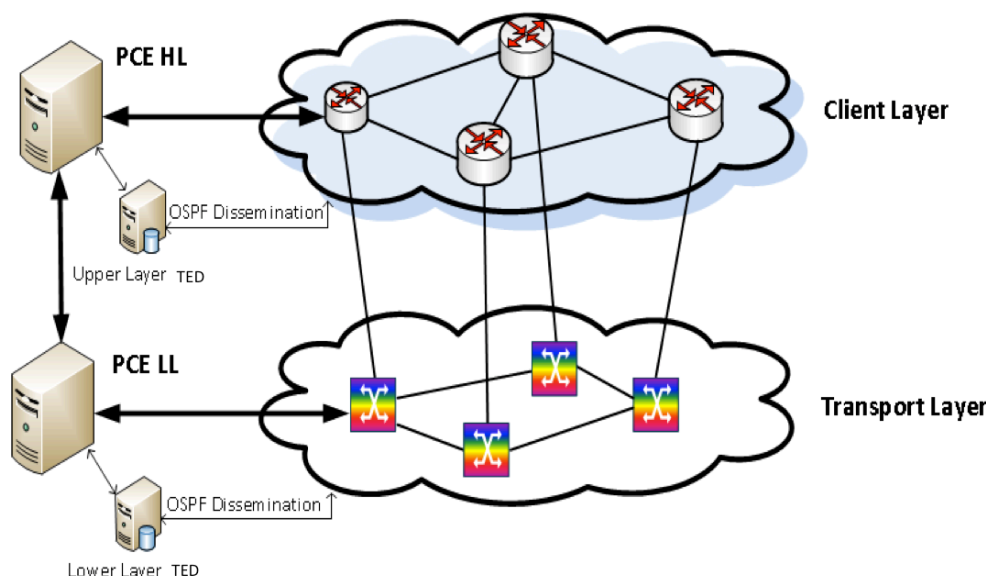


Figura 2-12: PCEs en capas separadas

En las situaciones con capas separadas como en la Figura 2.11, se puede dar que el camino óptimo desde el punto de vista de la capa superior no sea el óptimo desde el punto de vista de la capa inferior. Otro tipo de situaciones a tener en cuenta son las relacionadas con los fallos y las solicitudes de protección en caminos. Se podría dar el caso de que al solicitar un camino con protección disjunta en la capa superior, desde el punto de vista de ésta el camino sea disjunto (no comparta recursos de la capa superior) pero la capa inferior si los compartiera.

Con el objetivo de conseguir una optimización de red que tenga en cuenta las dos capas se define el concepto de PCE multicapa (ML-PCE) que obtiene información de las capas por separado y de los enlaces entre capas para ser capaz de realizar un cálculo óptimo de rutas multicapa.

El estándar actual [12] del ML-PCE lo presenta como elemento capaz de mejorar la eficiencia de la red completa proporcionando una interfaz estándar (PCEP) para permitir las solicitudes en entornos multicapa.

En la Figura 2-13 se presenta un caso típico de utilización del PCE multicapa en el que un nodo de la capa superior necesita establecer un camino hacia el nodo en el otro extremo de su misma capa. Si atendiera únicamente al plano de control de la capa superior, el camino elegido necesitaría de atravesar todos los nodos de la capa superior, por el contrario, en caso de haber recursos disponibles en la capa inferior, el ML-PCE podría sugerir un camino eficiente en recursos estableciendo una adyacencia nueva gracias a la capa inferior.

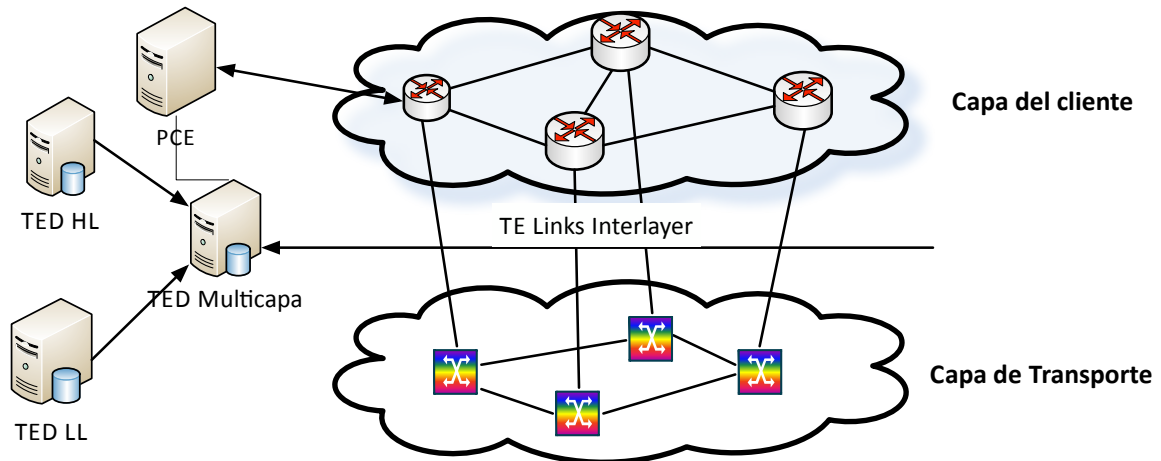


Figura 2-13: PCE Multicapa

El PCE multicapa necesita obtener la TED completa [41] de ambas capas y para ello debe por un lado conseguir la información de conexiones entre capas (TE-Links entre capas) y por otro las de las capas que se pueden ser proporcionadas por PCEs de capa o escuchando el plano de control de las mismas en caso de no existir PCE de capa. Una vez obtenida la información puede almacenarla en una TED completa con la información de la topología completa, o mantener una TED por cada capa. Este PFC evaluará ambas soluciones (PCE multicapa o separado) en términos de bloqueo y retardo.

2.5 VNTM

El modelo UNI ha sido presentado previamente en este proyecto como un mecanismo capaz de proveer conectividad y restauración para una red multicapa tanto en la capa MPLS como en la capa de transporte. Sin embargo, debido a limitaciones de este método, se propone un nuevo sistema técnico llamado *Virtual Network Topology Manager (VNTM)*. El VNTM sirve para ayudar en la gestión y la coordinación de las redes multicapa, permitiendo comunicación entre capas y una visión global de toda la red con las limitaciones de todos sus elementos. Puede también almacenar información de *Shared Risk Link Group (SRLG)* dada por las extensiones del UNI, lo que ayuda en la planificación y la protección cuando existen fallos en la red.

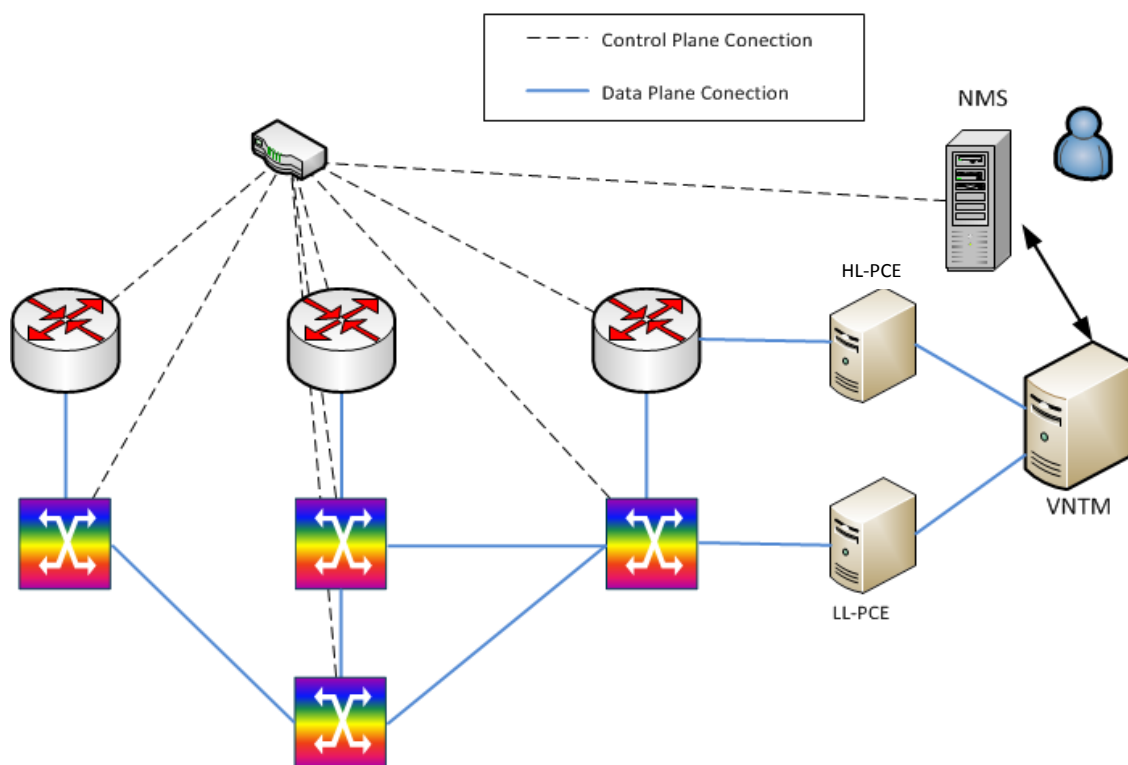


Figura 2-14: Esquema modelo de red multicapa con VNTM

Como podemos ver en esta imagen, la arquitectura propuesta para un dominio único está compuesta por dos PCEs, cada uno para una capa, el de IP/MPLS (HL-PCE) y el de transporte (LL-PCE), y un módulo VNTM. El VNTM no solo se encarga de mantener el estado de los links *inter-layer*, sino que también se encarga de calcular la capacidad máxima que podemos establecer entre dos nodos de la capa IP/MPLS con sus correspondientes LSPs establecidos en la capa de transporte. Para computar todos los posibles links entre cada par de nodos IP/MPLS, VNMT sigue el siguiente procedimiento:

1. Selecciona un camino sobre la capa de transporte entre los nodos ópticos correspondientes al par de nodos IP/MPLS. Esta camino se calcula mediante el LL-PCE (PCE que se encarga de la capa de transporte) con la menor granularidad.
2. En caso de que se encuentre un camino en la capa inferior se comunica al NMS con un mensaje afirmativo a través de una sesión PCEP.
3. El NMS se encarga de establecer el nuevo TE Link entre los dos nodos IP/MPLS con una cierta capacidad y posteriormente establece el LSP sobre este TE Link, ocupando el ancho de banda necesario para la petición.
4. VNTM vuelve a repetir el proceso cada vez que se recibe una petición para aumentar el ancho de banda en el TE Link.

Realizando este proceso, el VNTM es capaz de establecer los caminos multicapa para que el HL-PCE (PCE encargado de la capa IP/MPLS) pueda resolver caminos para las peticiones de conexión entre los pares de nodos de la capa IP/MPLS, también cuando no haya conectividad disponible en esa capa. Como ya hemos dicho gracias al establecimiento

de los caminos LSPs por parte del VNTM en la capa óptica, se crean los *light-paths* sobre los que rutamos las peticiones IP/MPLS.

El VNTM puede situarse en una única entidad (ver Figura 2-15) o distribuirse en la red por distintas entidades. En este último caso, hay dos posibles organizaciones del VNTM en la red: un VNTM padre que se encarga del resto, o todos los VNTMs con la misma relación (al mismo nivel).

La arquitectura puede ser usada también con múltiples dominios con múltiples PCEs. También, y es la manera más óptima, cuando tenemos un único PCE multicapa con visión global de toda la red.

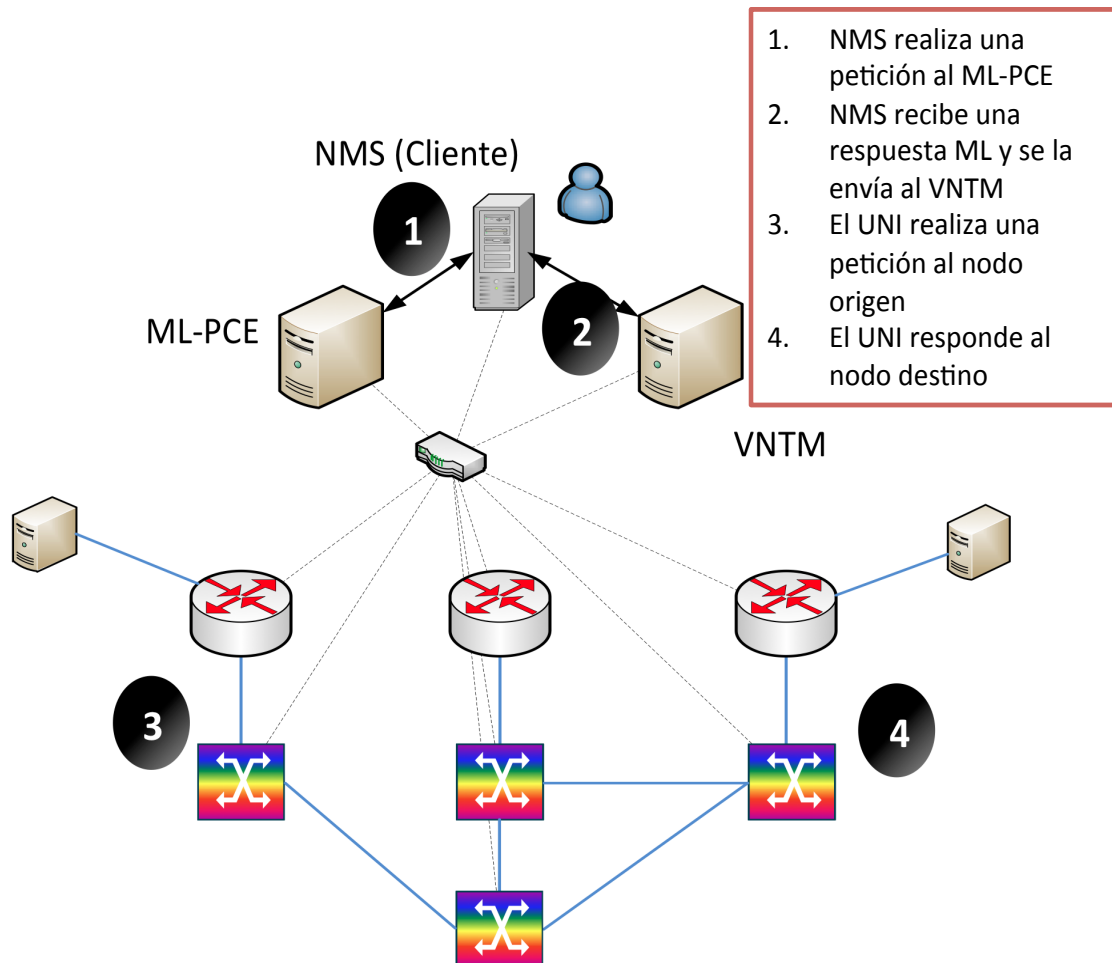


Figura 2-15: Esquema funcionamiento con VNTM

La parte de computación de caminos mediante los distintos algoritmos multicapa se detalla más adelante en la sección 5. Ahí se presentan los dos modelos de funcionamiento del PCE Multicapa: con TEDs separadas para cada capa, o con una única TED multicapa. El primer modelo sería el equivalente a tener 2 PCEs separados, uno para cada capa como podemos observar en la Figura 2.13. Mientras que el modelo de TED Multicapa corresponde al modelo de PCE Multicapa, como podemos ver en la Figura 2.14. Tanto para un modelo como para el otro el funcionamiento del NMS y del VNTM no varía en absoluto.

3 Definición de los escenarios de experimentación

Uno de los objetivos de este proyecto es la implementación de soluciones para el PCE. Para poder probar las implementaciones debemos disponer de escenarios (*testbeds*) de experimentación y hay que ajustarse a los requerimientos de cada uno para conseguir hacer funcionar las pruebas eficientemente. Durante la fase de experimentación de este proyecto se han utilizado dos *testbeds*: el de *Adrenaline* situado en el CTTC y el de *Telefónica I+D*. Para el *testbed* del CTTC se han desarrollado 5 algoritmos para redes WSON, mientras para el *testbed* de TID se han desarrollado tanto algoritmos de redes WSON como algoritmos de redes multicapa. En esta sección se van explicando las características de cada uno de ellos.

3.1 Testbed de Adrenaline

3.1.1 Descripción funcional

En esta arquitectura tenemos que utilizar dos servidores para los siguientes módulos [32]:

- El PCE es un proceso asíncrono multi-hilo que funciona como un servidor PCEP/TCP de forma que pueda aceptar y computar peticiones de los PCCs. En caso de tener un escenario multi-dominio el PCE puede actuar también como cliente enviando peticiones a otros PCEs de otros dominios.
- El proceso del PCC tiene un único hilo y funciona como un cliente PCEP/TCP para enviar peticiones de camino al correspondiente PCE.

Ambos funcionan sobre un sistema POSIX (GNU/Linux en particular).

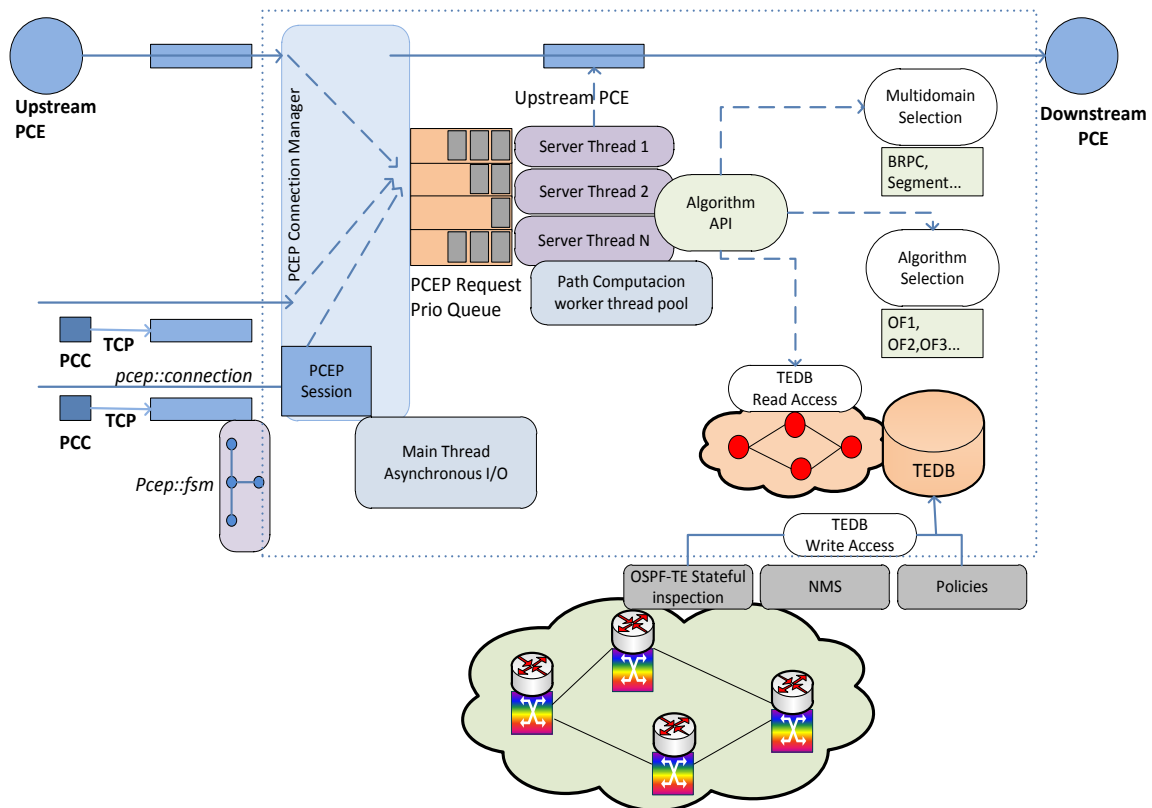


Figura 3-1: Arquitectura del PCE en el *testbed* de Adrenaline

El protocolo PCEP, junto con una máquina de estados (FSM) que tiene asociado, se encarga de dirigir el módulo *Connection Manager* que permite dar servicio concurrente a varios clientes. Una vez establecida la conexión el PCE a la petición con la ruta calculada, basándose en la TED almacenada en ese momento en la caché local. El PCE actúa como un lector de esta caché, lo que nos da dos tipos distintos de PCE: *stateless* o *stateful*. En el *stateless* PCE no se tienen en cuenta el conocimiento de computaciones de camino previas al calcular la ruta, mientras que en el *stateful* sí. El proceso del PCE también dedica algunos hilos para ir actualizando la TED en la caché. Estas modificaciones vienen dadas por los mensajes OSPF que denotan las nuevas características de los links o nodos de la red.

En este *testbed* la TED está modelada como una “*Graph Data Structure*”, es decir se ve como una instancia de un grafo con toda la información que conlleva.

3.1.2 Descripción del plano de control

En esta sección vamos a describir el plano de control GMPLS utilizado para el *testbed* de *Adrenaline* [32]. Actualmente este plano de control soporta una red WSON de dominio único, en la cual se ha colocado regeneradores 3R sobre una serie de nodos de la red. Estos regeneradores 3R se utilizan tanto para regenerar la señal óptica como para convertir (eléctricamente) la longitud de onda de cada canal. De esta manera cada conexión óptica establecida puede ser fijada sobre un solo *light-path* (ej. en camino óptico punto a punto) o sobre una serie de *light-path* concatenados. En estos nodos intermedios es donde encontramos los regeneradores 3R que se utilizan para que la calidad de transmisión no

disminuya por debajo de un cierto umbral y para controlar la restricción de continuidad de la longitud de onda. En este escenario, la finalidad es desarrollar una serie de algoritmos que dinámicamente sean capaces de recibir, encaminar conexiones ópticas y asignar longitudes de onda a los correspondientes segmentos de *light-path* calculados para la ruta. Para satisfacer la calidad de transmisión en toda la ruta, se han realizado extensiones en el estándar GMPLS OSPF-TE (protocolo de encaminamiento) de tal manera que se disemine información de impedimentos físicos, disponibilidad de los regeneradores 3R, disponibilidad de longitud de onda en cada canal y demás atributos de TE. De esta manera el PCE tiene una visión global del estado de la red y el estado de sus recursos (TED). Toda esta información la saca a partir de los mensajes OSPF-TE. En este escenario el indicador físico que tomamos para computar los niveles de calidad de señal óptica es el *Optical to Signal Noise Ratio (OSNR)*.

Las principales características del plano de control GMPLS en el *testbed* de *Adrenaline* son las siguientes:

- Un plano de control distribuido (*Unified Control Plane, UCP*) con 22 controladores GMPLS. Estos controladores son responsables de reservar las conexiones ópticas intermedias (ej. el protocolo RSVP-TE) y de la diseminación de recursos utilizados en la red, información que se va guardando en la TED de cada nodo.
- Un único *Path Computation Element (PCE)* situado en un controlador de conexión GMPLS, el cuál recibe peticiones de los PCCs y computa las rutas sobre la topología de red guardada en la TED a través de la diseminación de OSPF-TE.
- Permite el diseño, implementación, compilación automática y despliegue de algoritmos de cómputo de ruta para el PCE. En la API del CTTC se pueden encontrar todas las implementaciones y métodos necesarios para el desarrollo de los algoritmos deseados.
- Monitoreo básico del *testbed*.
- Una topología de red fija para todas las simulaciones. Esta topología se basa en la *STRONGEST European Optical Network*. Está formada por 22 nodos y 34 links bidireccionales que soportan 32 longitudes de onda por *link* y dirección, y 3 regeneradores por nodo (Figura 3–2).



Figura 3-2: European Optical Network

3.2 Testbed de Telefónica I+D

3.2.1 Descripción funcional

En este *testbed* realizamos pruebas de WSON y multicapa. Para el caso de WSON tenemos dos módulos que simulamos en dos servidores distintos:

- El PCE Server que funciona como un servidor TCP/PCEP de tal forma que recibe peticiones, computa caminos y envía respuestas al PCC.
- El PCC (cliente) funciona como un cliente TCP/PCEP que envía peticiones al PCE y recibe las respuestas. A la vez funciona como emulador de red, comprobando que los caminos computados por el PCE son factibles y se pueden establecer los LSPs sin problema. En caso de que se pueda establecer el LSP del camino computado, enviamos los mensajes OSPF de actualización de la TED de la topología. Una vez el PCE recibe los mensajes va actualizando su propia TED interna.

El emulador de red del cliente se encarga de comprobar que los recursos asignados a un camino en la computación del PCE no hayan sido previamente asignados a otra petición. En caso de que se dé concurrencia de recursos en caminos computados por el PCE, el cliente lo detecta y notifica la petición como “*Stolen Lambda*” [44]. Es decir, ocurre el fenómeno denominado como robo de lambda que se da cuando el PCE asigna a un camino una lambda que ya está en uso en algún link de la ruta. Si se da este caso tenemos que desocupar los enlaces previamente establecidos del camino, por lo que enviamos mensajes OSPF al PCE, liberando así los recursos ya ocupados. Este efecto también ocurre en el testbed de Adrenaline, el emulador de red debe tener en cuenta todos los efectos del plano de control como si fuese un plano de control completo.

Este fenómeno ocurre debido a los retardos que tenemos desde que enviamos una respuesta desde el PCE al PCC, hasta que nos llega la actualización OSPF de los enlaces que forman el camino. Debido a que hasta que no llegan estas actualizaciones en la TED del PCE esos recursos no figuran como utilizados, el PCE puede seguir asignándolos en las siguientes peticiones. Este retardo está compuesto por los retardos del plano de control para la reserva de recursos (establecimiento del LSP llevado a cabo por el emulador de red), el retardo de envío de la respuesta PCEP, y el retardo de envío de los mensajes OSPF de actualización de los recursos de red. Como podemos observar en la figura 3-3 el establecimiento del LSP por parte del emulador y los retardos en el envío de los mensajes entre el PCE y el PCC.

Para evitar que se produzca este problema tenemos que crear un mecanismo de pre-reserva los recursos que utilizamos al computar las rutas en el PCE. De forma que el algoritmo de encaminamiento que utilice el PCE tenga en cuenta la comprobación de reserva de recursos para saber los que están disponibles para ser usados.

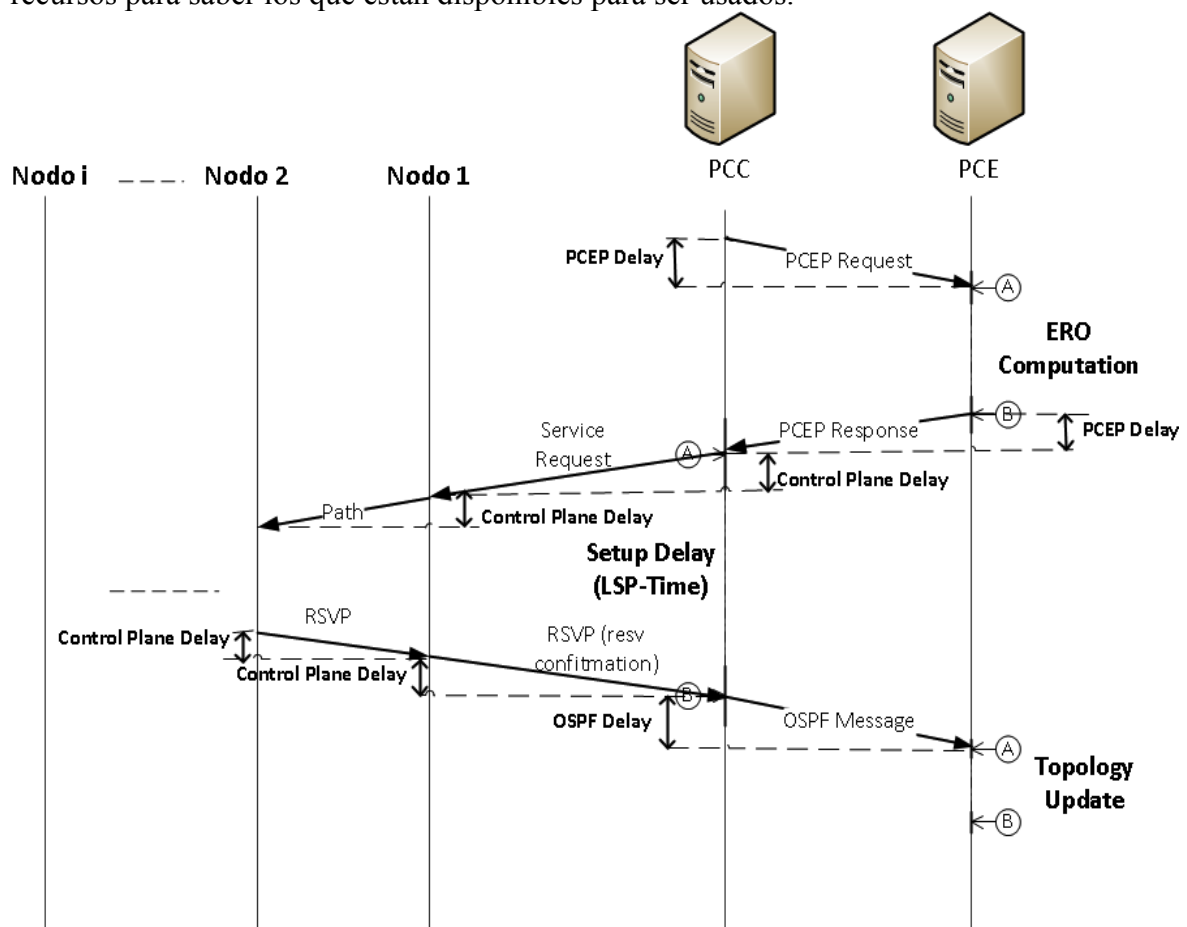


Figura 3-3: Diagrama de tiempos de funcionamiento del testbed

Para el modelo multicapa tenemos tres módulos que tenemos que simular en tres servidores. En nuestro caso el NMS y el VNTM los simularemos en un mismo servidor asignando puertos distintos para que no haya problema:

- El PCE Server que funciona como un servidor TCP/PCEP de tal forma que recibe peticiones, computa caminos y envía respuestas al PCC.
- El PCC (cliente) funciona como un cliente TCP/PCEP que envía peticiones al PCE y recibe las respuestas. A la vez funciona como emulador de red para la capa

IP/MPLS (capa del cliente), comprobando que los caminos IP/MPLS computados por el PCE son factibles y se pueden establecer los LSPs sin problema. En caso de que se pueda establecer el LSP del camino computado, enviamos los mensajes OSPF de actualización de la topología IP/MPLS. Una vez el PCE recibe los mensajes va actualizando su propia TED interna. Si la respuesta del PCE conlleva el establecimiento de un nuevo *light-path* el NMS le pasa la respuesta al VNTM. Si el LSP no puede establecerse correctamente indica el robo de recursos correspondiente, en este caso ancho de banda del *light-path*.

- El VNTM funciona como servidor TCP/PCEP. Recibe las respuestas que le pasa el NMS después de haber sido procesadas. Posee un emulador de red para la capa óptica (capa de transporte) y se encarga de comprobar que los caminos ópticos computados por el PCE son factibles y se pueden establecer los LSPs sobre ellos para dar soporte a los futuros *light-paths*. Si ocurre algún error o el link sobre el que queremos establecer el LSP está ocupado ya, indica el robo del recurso que queremos utilizar (en este caso *lambdas*). A continuación, en la siguiente subsección se explica más detalladamente el papel de este módulo.

3.2.2 Descripción del plano de control

En la siguiente figura 3.4 podemos observar el esquema de funcionamiento de los 3 módulos con todas las conexiones que existen entre ellos en el plano de control multicapa:

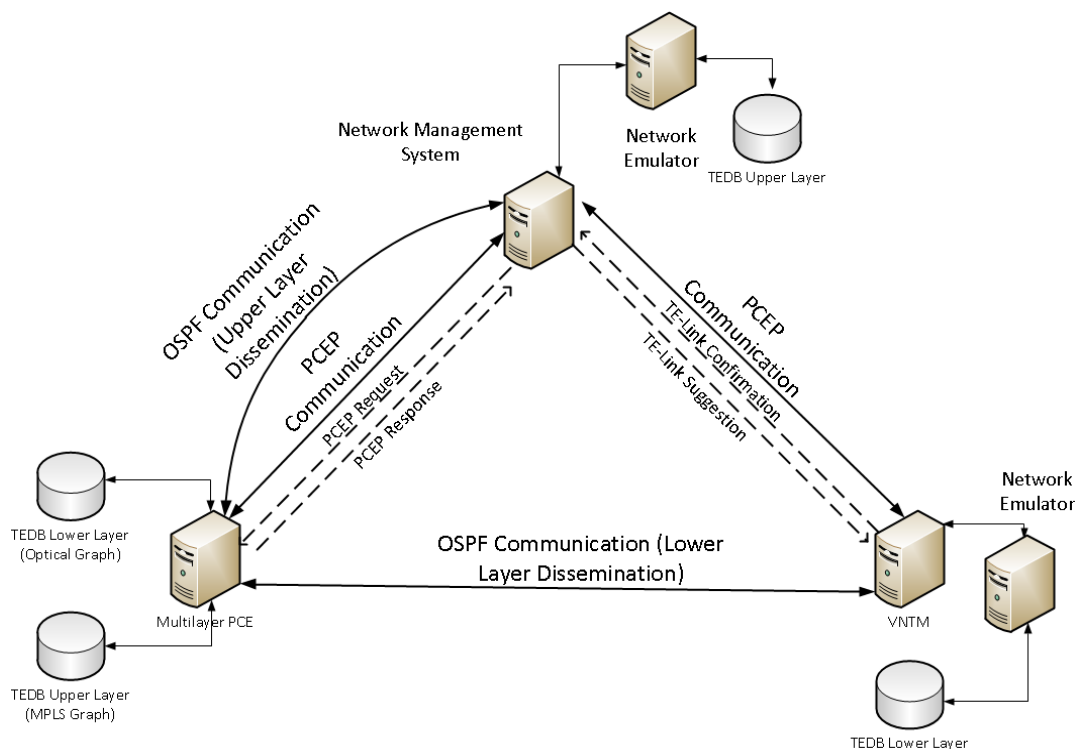


Figura 3-4: Esquema modelo de Plano de Control de TID

En la siguiente captura de *wireshark* (Figura 3.5) podemos observar los mensajes de establecimiento de conexión PCEP entre el NMS y el PCE, y a continuación 2 mensajes de petición de camino con sus correspondientes respuestas del PCE.

Io.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.1.71	172.16.1.70	TCP	74	35387 > pcep [SYN] Seq=0 win=14600 L
2	0.000038	172.16.1.70	172.16.1.71	TCP	74	pcep > 35387 [SYN, ACK] Seq=0 Ack=1
3	0.005465	172.16.1.71	172.16.1.70	TCP	66	35387 > pcep [ACK] Seq=1 Ack=1 win=1
4	0.024332	172.16.1.70	172.16.1.71	PCEP	78	OPEN MESSAGE
5	0.058848	172.16.1.71	172.16.1.70	PCEP	78	OPEN MESSAGE
6	0.059443	172.16.1.70	172.16.1.71	PCEP	70	KEEPALIVE MESSAGE
7	0.090251	172.16.1.71	172.16.1.70	PCEP	70	KEEPALIVE MESSAGE
8	3.551317	172.16.1.71	172.16.1.70	PCEP	110	PATH COMPUTATION REQUEST MESSAGE
9	3.611136	172.16.1.70	172.16.1.71	PCEP	110	PATH COMPUTATION REPLY MESSAGE
10	7.181019	172.16.1.71	172.16.1.70	PCEP	110	PATH COMPUTATION REQUEST MESSAGE
11	7.185432	172.16.1.70	172.16.1.71	PCEP	178	PATH COMPUTATION REPLY MESSAGE
12	30.109627	172.16.1.70	172.16.1.71	PCEP	70	KEEPALIVE MESSAGE
13	30.141399	172.16.1.71	172.16.1.70	PCEP	70	KEEPALIVE MESSAGE

Figura 3-5: Mensajes capturados con *wireshark*

Los mensajes de petición de camino (PATH COMPUTATION REQUEST MESSAGE) se caracterizan por contener los siguientes campos que podemos contemplar en la captura de la siguiente imagen:

- [-] Path Computation Element communication Protocol
 - [+] PATH COMPUTATION REQUEST MESSAGE Header
 - [+] RP object
 - [-] END-POINT object
 - object Class: END-POINT OBJECT (4)
 - object Type: 1
 - [+] Flags
 - object Length: 12
 - Source IPv4 Address: 192.168.8.3
 - Destination IPv4 Address: 192.168.8.1
 - [-] BANDWIDTH object
 - object Class: BANDWIDTH OBJECT (5)
 - object Type: 1
 - [+] Flags
 - object Length: 8
 - Bandwidth: 4,000000
 - [-] OBJECTIVE FUNCTION object
 - object Class: OBJECTIVE FUNCTION OBJECT (OF) (21)
 - object Type: 1
 - [+] Flags
 - object Length: 8
 - OF-Code: Unknown (1100)

Figura 3-6: Mensaje de petición PCEP

El PCE puede devolvernos 3 tipos de ERO distintos:

1. ERO multicapa: camino establecido sobre la capa de transporte (óptica).

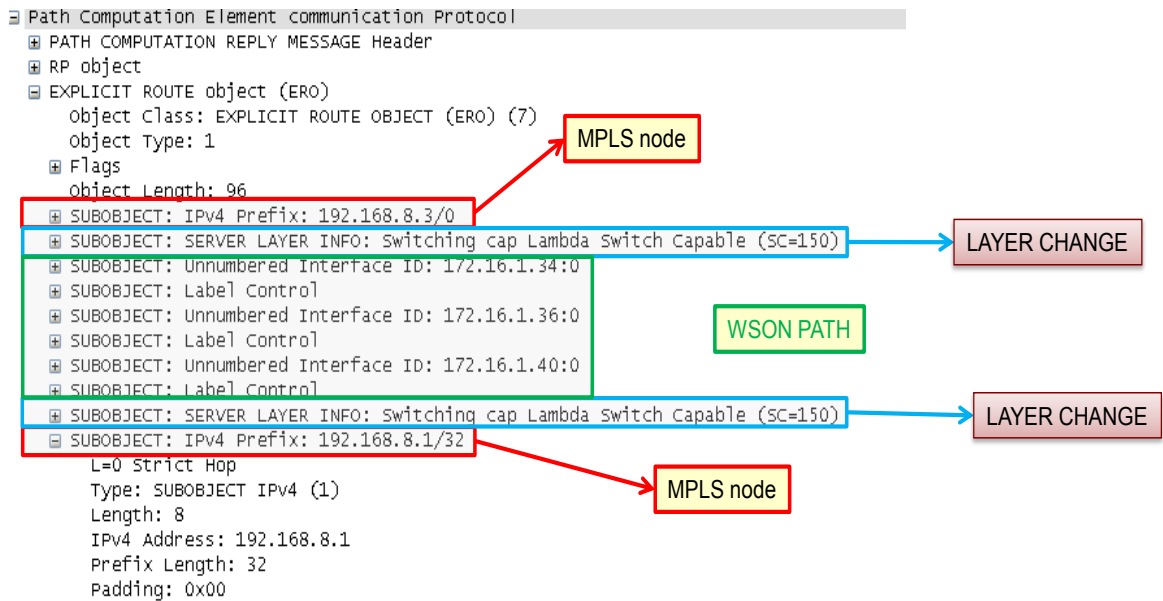


Figura 3-7: Respuesta ERO Multicapa

Como podemos observar este ERO contiene los nodos origen y destino de la capa IP/MPLS y los nodos ópticos de la capa de transporte. Mediante el sub-objeto `SERVER_LAYER_INFO` delimitamos los cambios de capa. Los sub-objetos *Label Control* contienen la lambda seleccionada para el cómputo del camino óptico. Este es el caso de la Figura 3.7.

2. ERO IP/MPLS: camino establecido exclusivamente sobre los *light-paths*, TE Links establecidos en la capa del cliente (IP/MPLS).

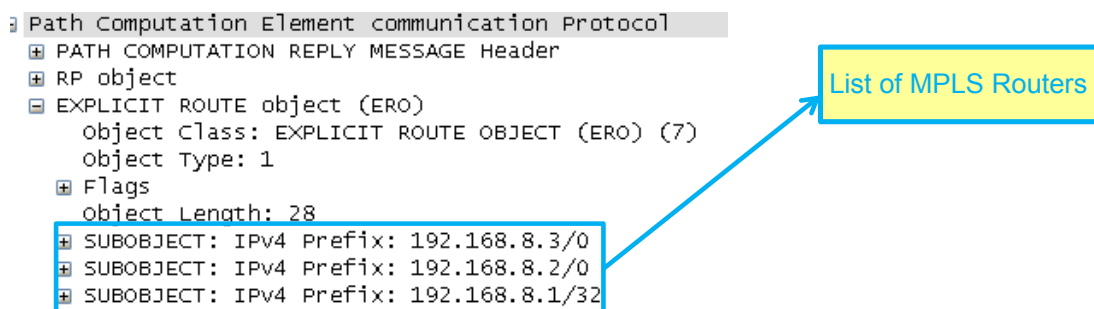


Figura 3-8: Respuesta ERO IP/MPLS

3. ERO mixto: con partes de camino establecidas sobre *light-paths* ya creados previamente, y partes sobre la capa de óptica. Mediante el sub-objeto `SERVER_LAYER_INFO` delimitamos los cambios de capa. El ERO resultante sería una mezcla de los dos EROs mostrados previamente.

Cuando al NMS llega una respuesta del PCE lo primero es analizar si el ERO lleva el sub-objeto `SERVER_LAYER_INFO`. En caso de encontrar este sub-objeto es indicación de que

hay partes de camino que transcurren sobre diferentes capas (véase la Figura 3.9). Por ello tenemos que separar las partes de la capa óptica para enviárselas al VNTM a través del mensaje TE_LINK_SUGGESTION. Si no encontramos este sub-objeto en el ERO esto nos indica que el camino de respuesta solo transcurre por la capa superior (véase la Figura 3.10).

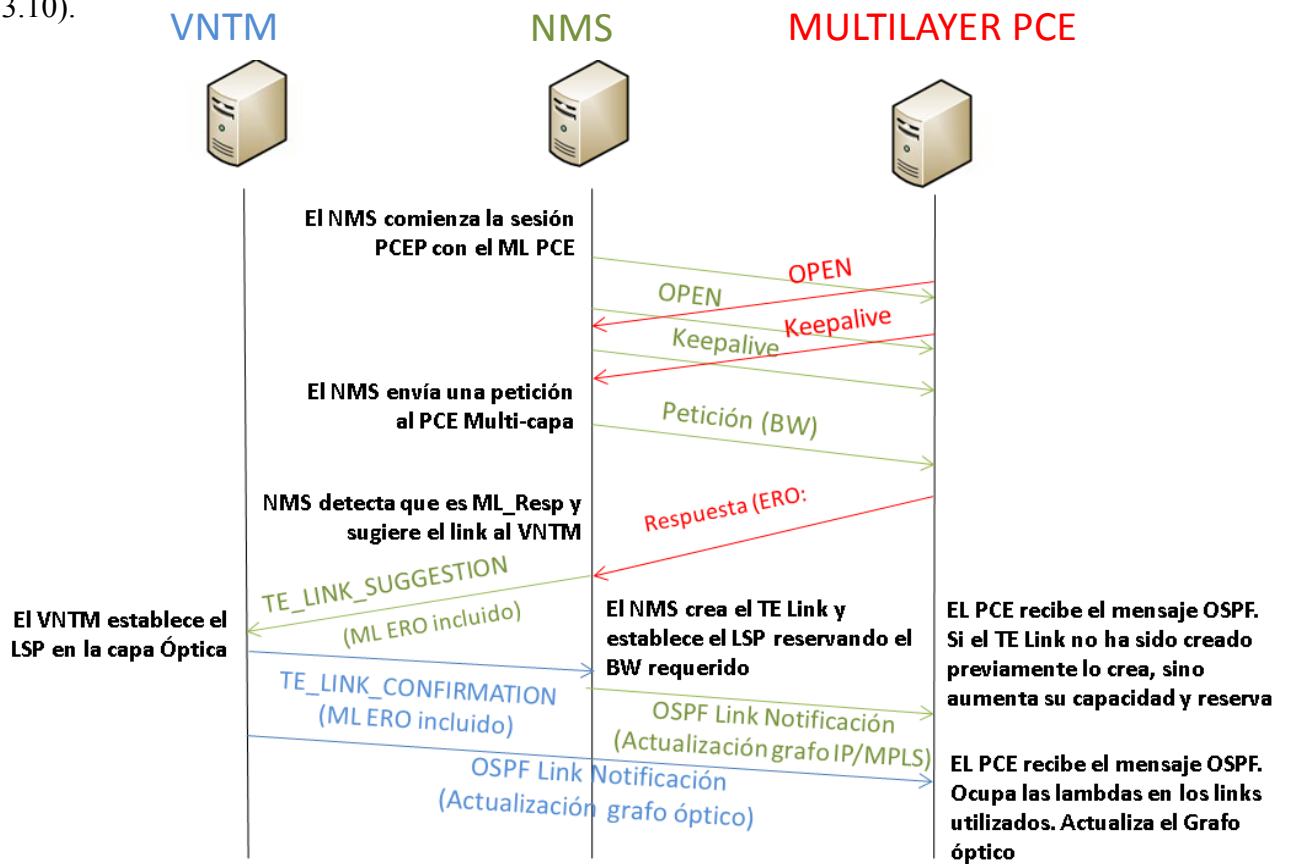


Figura 3-9: Diagrama de intercambio de mensajes para Respuesta ML

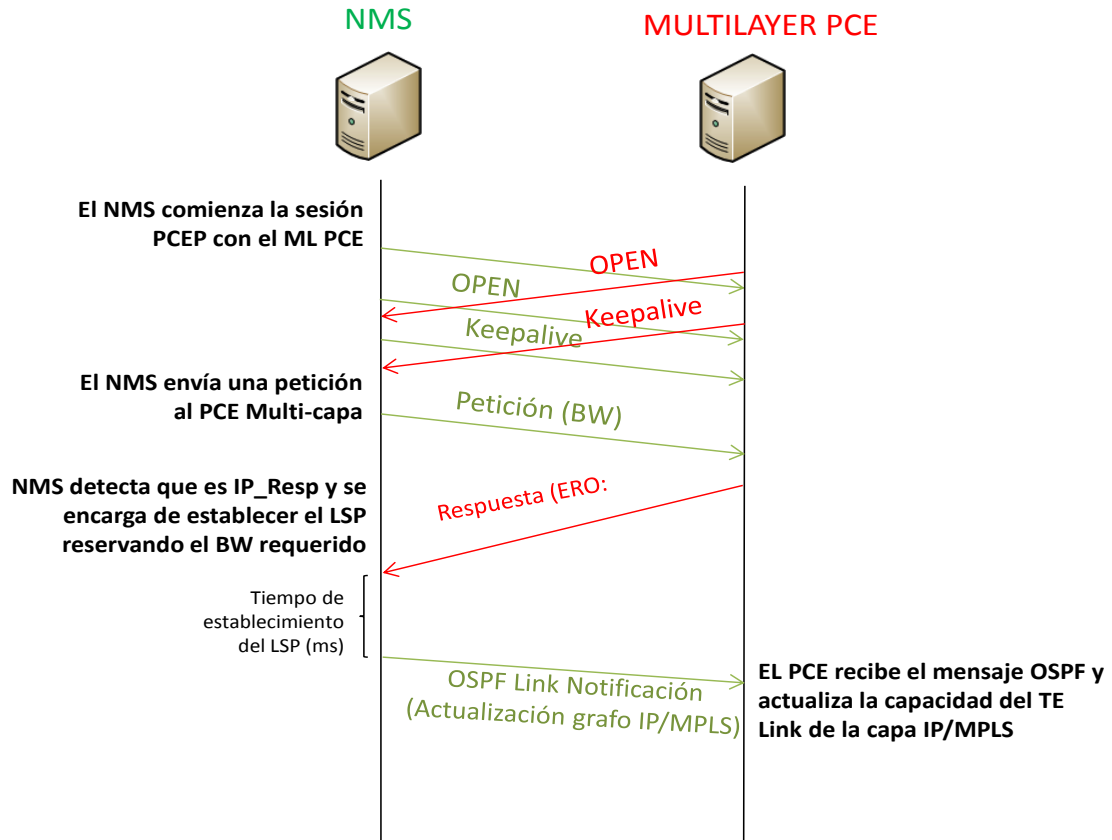


Figura 3-10: Diagrama de intercambio de mensajes en caso de respuesta IP/MPLS

Para estudiar el rendimiento de los algoritmos multicapa utilizamos dos topologías de red con distintas características. En primer lugar tenemos la topología de la red NSF, que tiene 14 nodos y 21 links, como podemos ver en la figura 3-11. Todos los nodos tienen su correspondiente nodo IP/MPLS en la capa superior, y el nodo óptico (OXC) en la capa de transporte. Cada link es bidireccional y lleva 32 *lambdas*. La capacidad que nos aporta cada *lambda* es OC-192 (10Gbps). No tenemos conversores de *lambda* en los nodos.

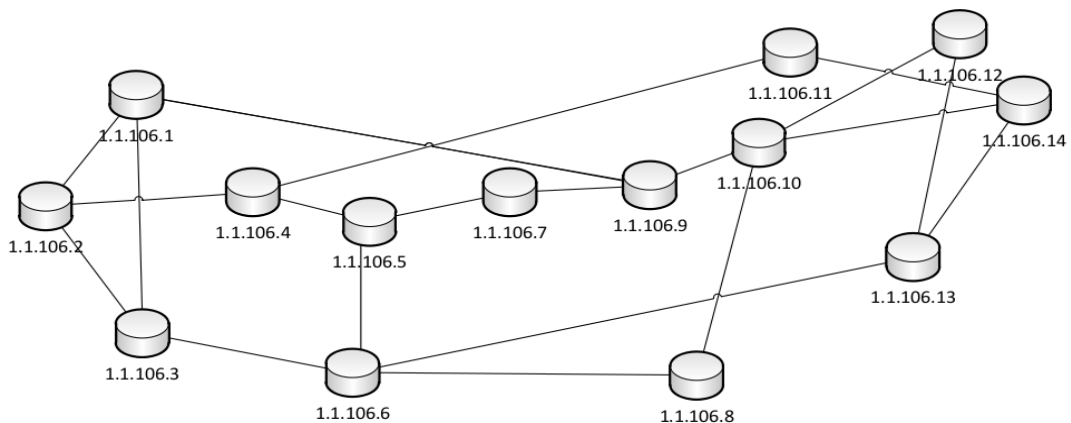


Figura 3-11: Topología de la NSF Network de EE.UU.

La otra topología que utilizamos para las simulaciones es la de la red de telefónica España. Está compuesta por 30 nodos y 56 enlaces de fibra óptica, como podemos observar en la figura 3-12:

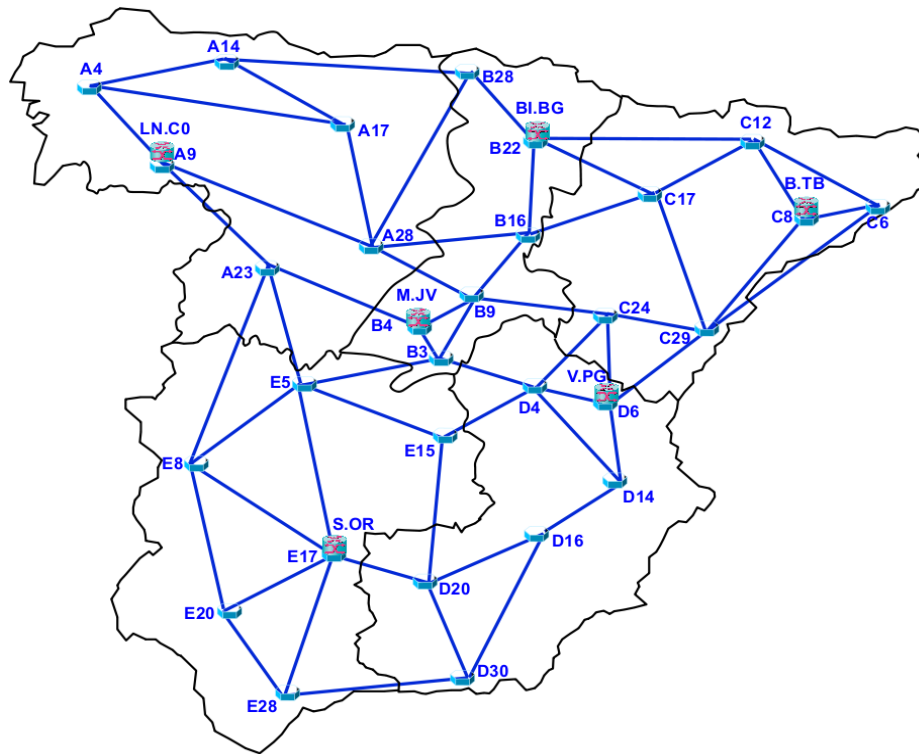


Figura 3-12: Topología Red de Telefónica España

4 Evaluación de algoritmos para redes de conmutación de longitudes de onda

Para la elaboración de estos algoritmos vamos a considerar el encaminamiento y la asignación de longitudes de onda en redes ópticas con conmutación de circuitos. Como vemos la solución al encaminamiento de conexiones en este tipo de redes está compuesto por dos sub-problemas:

1. Determinar un conjunto de caminos óptimos físicos para albergar la conexión.
2. Después buscar y seleccionar una λ óptima para el camino físico.

Tradicionalmente estas dos operaciones se han realizado de manera disjunta, sin tener en cuenta el estado de la red. En este caso vamos a tener en cuenta todos los posibles caminos entre los nodos de origen y destino, y posteriormente incorporaremos el estado de uso de la red actual para seleccionar un camino junto con una longitud de onda.

Dentro de los algoritmos de encaminamiento podemos encontrar dos tipos: adaptativos y estáticos. En el caso de los estáticos el procedimiento de encaminamiento no varía con el tiempo. Sin embargo los adaptativos usan información del estado de la red en el momento de establecer una conexión.

4.1 Definición de algoritmos implementados

Aquí presentamos los distintos algoritmos de encaminamiento que han sido implementados [31]. Cada uno de ellos hay que combinarlo posteriormente con los 5 criterios de selección de longitud de onda (λ) que se presentan en la sección 4.2.

4.1.1 KSP con Pre-cómputo (*Fixed Routing*)

Es un ejemplo de encaminamiento estático, donde cada par de nodos origen/destino tiene asignado un único camino. Una conexión se bloquea en caso de que su camino asociado no esté disponible en algún tramo. En este caso aplicamos *K-Shortest-Path* por lo que vamos a tener un conjunto de caminos para cada par origen/destino. En caso de que uno no esté disponible iremos comprobando los siguientes, y la conexión se bloqueará si ninguno de los caminos prefijados está disponible.

4.1.2 AURE

Como podemos observar en la Figura 4-1 este algoritmo mantiene la información de cada longitud de onda creando un grafo exclusivo para cada una. De forma que cuando ocupamos una λ de un link, el algoritmo se encarga posteriormente de borrar los enlaces del grafo correspondiente a esa λ hasta que deja de ser usada. Este algoritmo computa caminos para todas las λ s disponibles, es decir aplica *K-Shortest-Path* para tantos grafos como λ s tenemos, y posteriormente aplica el proceso de selección de λ (ver sección 4.2). En este caso se aplica la selección de λ exhaustiva (ver sección 4.2.4).

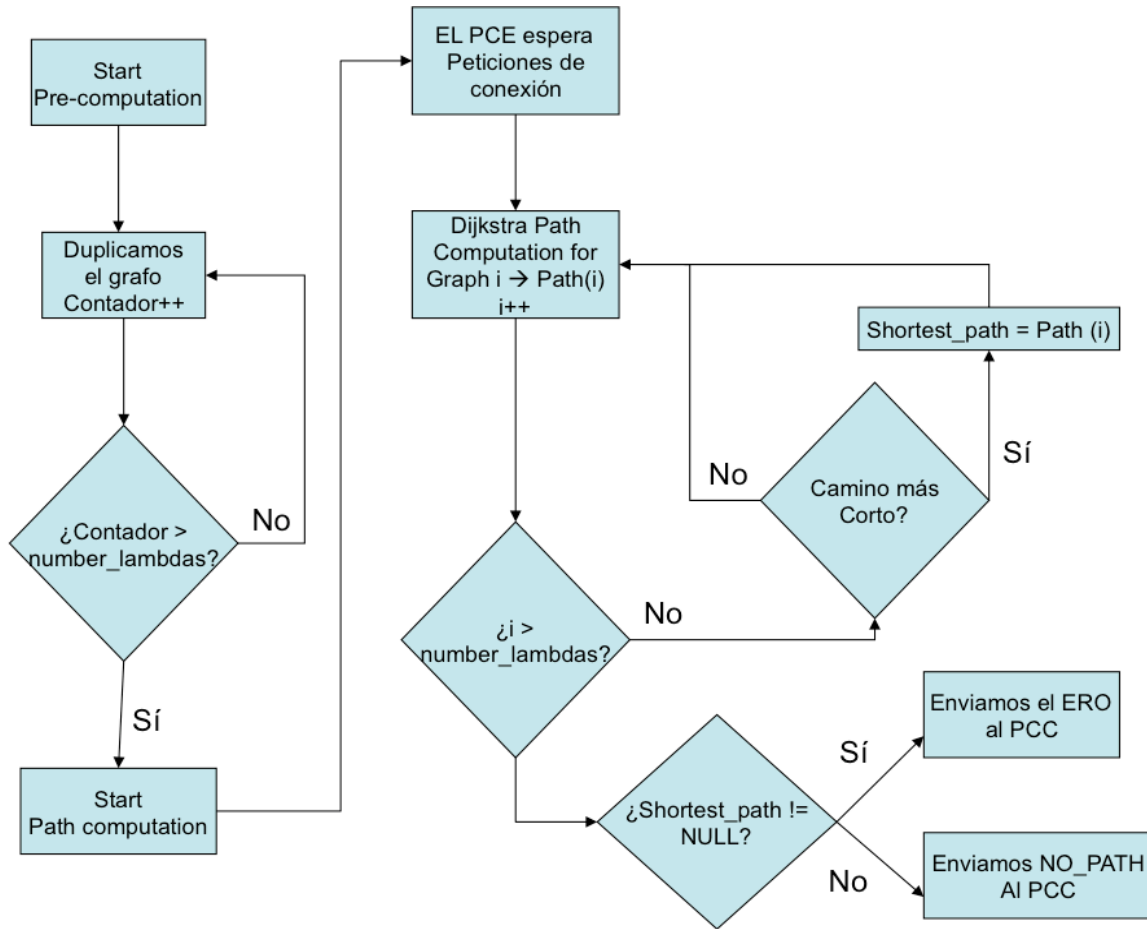


Figura 4-1: Esquema de funcionamiento del AURE

4.1.3 KSP cada conexión (Live)

Este sería un ejemplo de algoritmo adaptativo. En este caso aplicamos *K-Shortest-Path* de manera dinámica para cada conexión. Como podemos observar el hecho de realizar una computación de este algoritmo por cada petición hace que los tiempos aumenten considerablemente y, en el caso de no tener un mecanismo de pre reserva, esto hace que el robo de lambda aumente de forma notoria. Una vez tenemos la ruta tenemos que aplicar un criterio de selección de lambda.

4.2 Criterio de selección de longitud de onda

Vamos a investigar cinco tipos de algoritmos RWA considerando cinco distintos mecanismos de selección de longitud de onda. Los dos primeros que mostramos hacen uso del vector de utilización de longitudes de onda o lambda [31].

En nuestra red con K enlaces y W longitudes de onda, el estado de un link i , dónde $0 < i < K-1$ en un instante de tiempo t , puede ser definido por el vector columna:

$$\sigma_t^{(i)} = (\sigma_t^{(i)}(0), \sigma_t^{(i)}(1), \dots, \sigma_t^{(i)}(W-1))^T, \text{ donde } \sigma_t^{(i)}(j) = 1 \text{ si } \lambda_j \text{ está siendo}$$

utilizada por alguna conexión en el instante de tiempo t y $\sigma_t^{(i)}(j) = 0$ en caso contrario. Por lo tanto el estado de la red en un instante de tiempo t viene descrito por la matriz:

$$\sigma_t = (\sigma_t^{(0)}, \sigma_t^{(1)}, \dots, \sigma_t^{(K-1)})$$

Cuando llega una conexión en un instante de tiempo t el algoritmo de RWA trata primero de buscar un camino óptico $P = (i_1, i_2, \dots, i_l)$ desde el nodo origen al nodo destino de tal forma que se cumpla que $\sigma_t^{(i_k)}(j)$ para todo $k = 1, 2, \dots, l$ y algunos j . Donde j es la longitud de onda que estamos comprobando: $j, 0 < j < W$.

Gracias a la matriz de estado de la red podemos saber rápidamente qué longitudes de onda están siendo usadas en todos los enlaces de la red. Los dos primeros algoritmos hacen uso de la matriz de estado de la red. Por esta razón son los dos algoritmos que se utilizan para tratar de ver si este esfuerzo es realmente necesario para mejorar el rendimiento del bloqueo de la red. Una vez tenemos la selección de lambda realizada, hay que comprobar que esa longitud de onda sea factible y este libre en toda la ruta previamente seleccionada. A continuación detallan los algoritmos con los criterios que aplica cada uno para el encaminamiento [31]. Para más detalle se incluye una topología de red simple de cinco nodos y con cuatro longitudes de onda bidireccionales por cada enlace (ver Figura 4-2). Posteriormente para cada algoritmo se detalla la longitud de onda que sería seleccionada en el caso de querer establecer un camino del nodo 1 al nodo 3, el cual transcurriría por los nodos 1, 2 y 3:

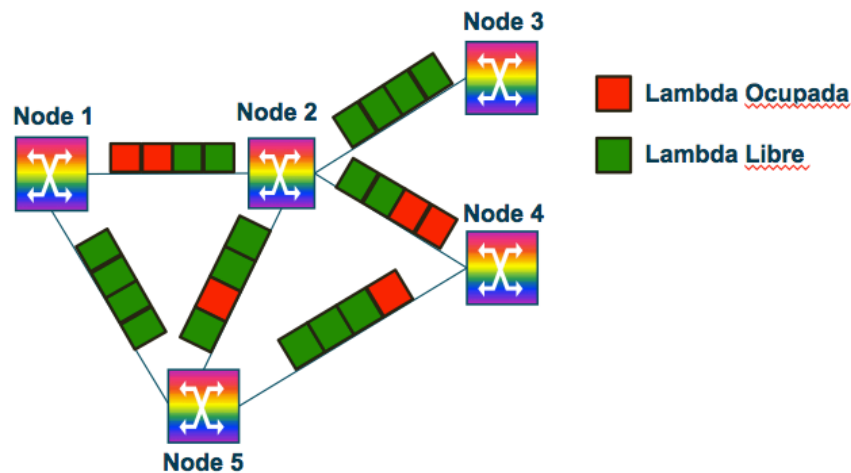


Figura 4-2: Red óptica: Ejemplo demostración selección de lambda

Vamos a resumir la utilización de cada lambda en la siguiente matriz de uso:

	Lambda 0	Lambda 1	Lambda 2	Lambda 3
Número de veces Ocupada	1	2	1	2

Tabla 4-1: Tabla de matriz de utilización actual de cada longitud de onda en la red

4.2.1 Pack

Este algoritmo intenta encaminar la conexión utilizando la longitud de onda más utilizada como criterio de selección. Las longitudes de onda son buscadas de forma descendente según su utilización actual en la red, de esta manera conseguimos maximizar el uso de las longitudes de onda disponibles.

Para el caso del ejemplo propuesto tenemos disponibles las lambdas 2 y la 3. Viendo en la tabla 4-1 vemos cómo la lambda más usada es la 3, por lo que es la que escogemos.

4.2.2 Spread

Este algoritmo intenta encaminar la conexión sobre la longitud de onda de menos utilización en la red. Es decir, las longitudes de onda se buscan en orden ascendente según su utilización actual en la red. De esta manera podemos decir que nos acercamos a una distribución uniforme de la carga sobre el conjunto de longitudes de onda.

Para este caso tenemos de nuevo las lambdas 2 y 3 disponibles en todo el camino. En este caso utilizamos la lambda 2, ya que es la menos utilizada a nivel global en la red.

4.2.3 Random

Este algoritmo selecciona la longitud de onda del conjunto de longitudes de onda disponibles de una manera aleatoria. Realizamos la búsqueda con una distribución uniforme sobre el conjunto de longitudes de onda.

En este caso para seleccionar la lambda en el ejemplo propuesto elegiríamos de manera aleatoria en las lambdas 0, 2 y 3.

4.2.4 Exhaustive

En este caso se realiza una búsqueda exhaustiva de computación del camino más corto para todas las longitudes de onda disponibles en la red. Posteriormente se selecciona el camino de métrica más corta entre todos los caminos computados, y se asigna la longitud de onda que correspondía a ese camino.

Para el ejemplo propuesto buscamos tantos caminos posibles como haya para llegar al nodo 3 con alguna lambda disponible. El proceso es el siguiente:

1. Empezando por la lambda 0 encontramos camino pasando por los nodos 1, 5, 2 y 3 (en este mismo orden). Como no tenemos otro camino mantenemos este como el más corto. En cada computación hay que eliminar del grafo de red todos los enlaces en los que la lambda sobre la que computamos esté ocupada.
2. Hacemos lo mismo con las siguientes lambdas, hasta que encontremos otro camino más corto aún.
3. Una vez hecha la computación con todas las lambdas nos quedamos con el camino que transcurre por la lambda 2, el que pasa por los nodos 1, 2 y 3. Este camino es el más corto obtenido computando con todas las lambdas.

4.2.5 Fixed (*First Fit*)

En este caso el orden de búsqueda de la longitud de onda es prefijado de ante mano. En nuestras pruebas vamos a utilizar *First Fit (FF)*, es decir, la primera longitud de onda disponible que veamos será sobre la que rutemos nuestra conexión.

En este caso utilizaríamos la primera lambda disponible en el camino elegido, es decir la lambda 0.

4.3 Efecto del Bloqueo de conexión por “robo de lambda”

El PCE computa caminos basándose en la información de estado de la red, la cuál esta almacenada en la TED. Como ya hemos comentado previamente, la TED se llena usando la información de los recursos de la red mediante el protocolo OSPF y sus extensiones de TE. Los elementos de red envían periódicamente mensajes OSPF como consecuencia de los cambios en la red que el PCE utiliza para mantener actualizada su TED. El problema es que esta actualización no es instantánea, sino que al ser un proceso dinámico del plano de control (ver figura 3-3) no encontramos con que durante un pequeño retardo de tiempo la TED del PCE se encuentra desactualizada. De tal manera que si nos llegan peticiones de camino durante ese pequeño período de tiempo, puede ocurrir que el PCE responda con recursos que ya están asignados para otro camino. Cuando la respuesta llega al plano de control, éste detecta que los recursos ya han sido ocupados por otro elemento de red, lo que causa el problema conocido como “robo de lambda”.

Para resolver este problema se propone un mecanismo de pre-reserva (PR). El mecanismo trata de añadir al PCE información sobre lo que ha ocurrido en previas computaciones del PCE, de tal manera que se pre-reservan los recursos en la TED del PCE durante un tiempo de reserva determinado (*Tresv*). Una vez el temporizador *Tresv* expira los recursos quedan libres y se reservan de nuevo para las próximas computaciones de camino. Se ha definido un tiempo de 3 segundos para el mecanismo de PreReserva.

Este mecanismo ha sido aplicado para las simulaciones en el *testbed* de Telefónica I+D. Por ello podemos ver que obtenemos resultados para dos casos: simulaciones sin pre-reserva en el PCE y simulaciones con pre-reserva. Como ya explicaremos más adelante, la pre-reserva nos permite aumentar cuantiosamente la carga que introducimos a la red, eliminando el bloqueo de conexión por “robo de lambda” y obteniendo bloqueo por falta de recursos (restricciones físicas).

4.4 Evaluación de los resultados

En esta sección se muestran los resultados obtenidos tanto en el caso de el *testbed* del CTTC como en el de TID. Podemos ver los rendimientos en cuanto a bloqueo de paquetes así como los tiempos de computación y de establecimiento de los LSPs.

4.4.1 Resultados CTTC

4.4.1.1 Probabilidad de bloqueo

En el CTTC se han implementado algoritmos de encaminamiento KSP (*Dijkstra*) combinándolo con los cinco criterios de selección de onda. Para ello se ha utilizado un tiempo de servicio exponencial de media 6 segundos y se ha ido variando el tiempo entre llegadas para conseguir ir variando la carga que le introducimos a la red. El número de muestras utilizado es de 10.000 conexiones y la topología utilizada es la de la Figura 3-2. Se han obtenido los siguientes resultados de rendimiento en cuanto a probabilidad de conexiones bloqueadas.

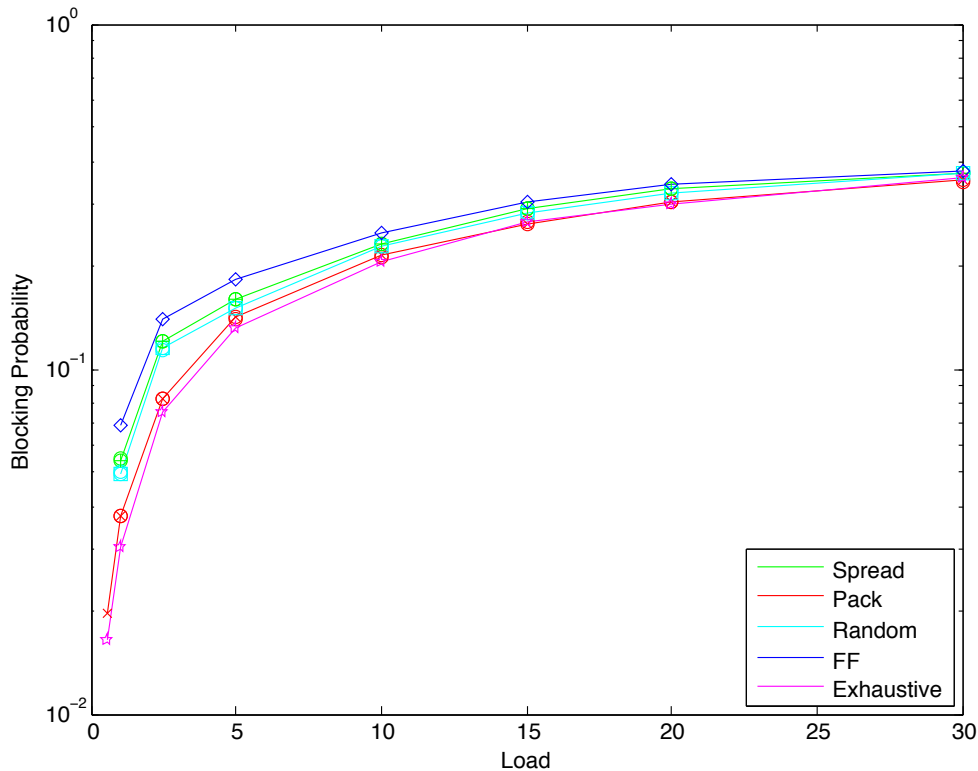


Figura 4-3: Probabilidades de bloqueo variando la carga de la red en el CTTC

Como podemos ver observar en los resultados la política de selección de lambda que mejores resultados aporta es la “Exhaustive”. Lo cuál es bastante lógico ya que según la teoría [31] es el más ordenado computacional y el que mejor hace uso de los recursos de la red. Si nos fijamos en los resultado que se obtienen en [31], podemos asegurar que quedan validadas las políticas implementadas para este *testbed*.

4.4.1.2 Tiempos de ejecución

Para estimar los tiempos de funcionamiento en el *testbed* del CTTC se muestran los resultados obtenidos con el algoritmo KSP-PACK con una carga de la red igual a 1 y 10.000 muestras (conexiones). En la Figura 4-4 se muestra el histograma de tiempos de computación de los caminos obtenidos en el PCE. La media de los tiempos de computación es de 51,7571 (ms). Este tiempo de computación consiste en el tiempo que pasa desde que el PCE recibe una petición de camino del PCC hasta que computa este camino y devuelve la respuesta con el ERO al PCC. Cuanto más complicado sea el algoritmo y más tiempo computacional requiera el algoritmo utilizado mayor será este tiempo.

En la Figura 4-4 podemos ver el histograma de tiempos de computación. Éste nos indica la cantidad de peticiones que se encuentran contenidas en cada rango de tiempos de computación, por lo que podemos ver que la mayoría de peticiones se encuentran en torno a 40-50 ms.

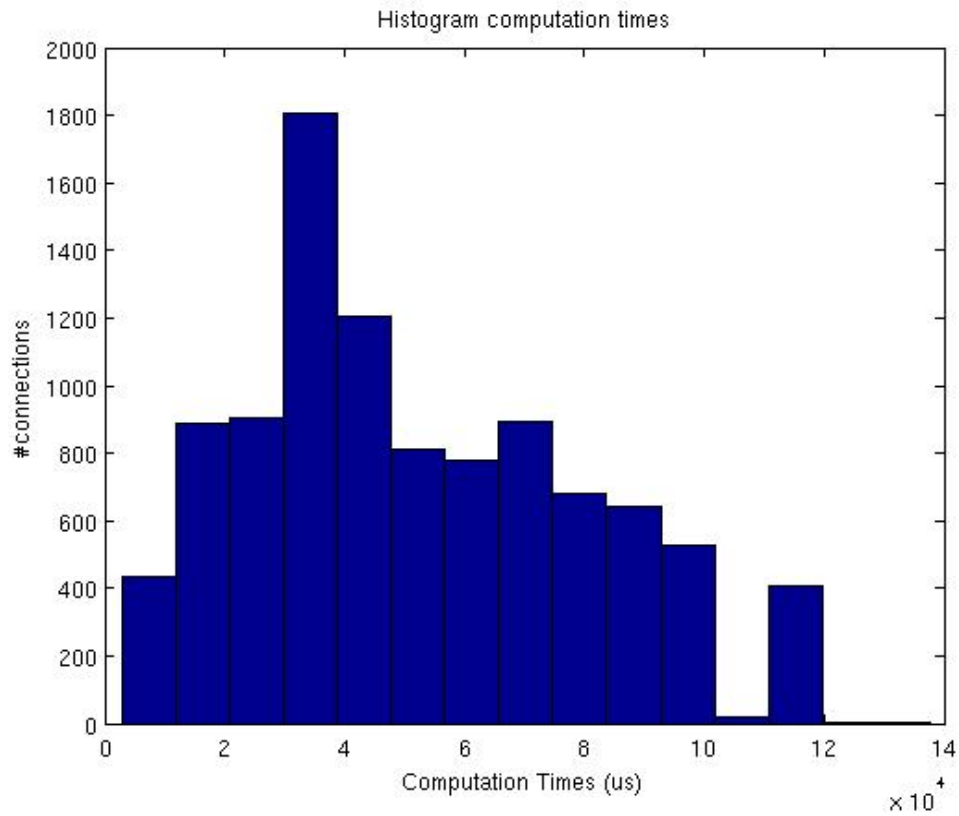


Figura 4-4: Histograma de tiempos de computación CTTC

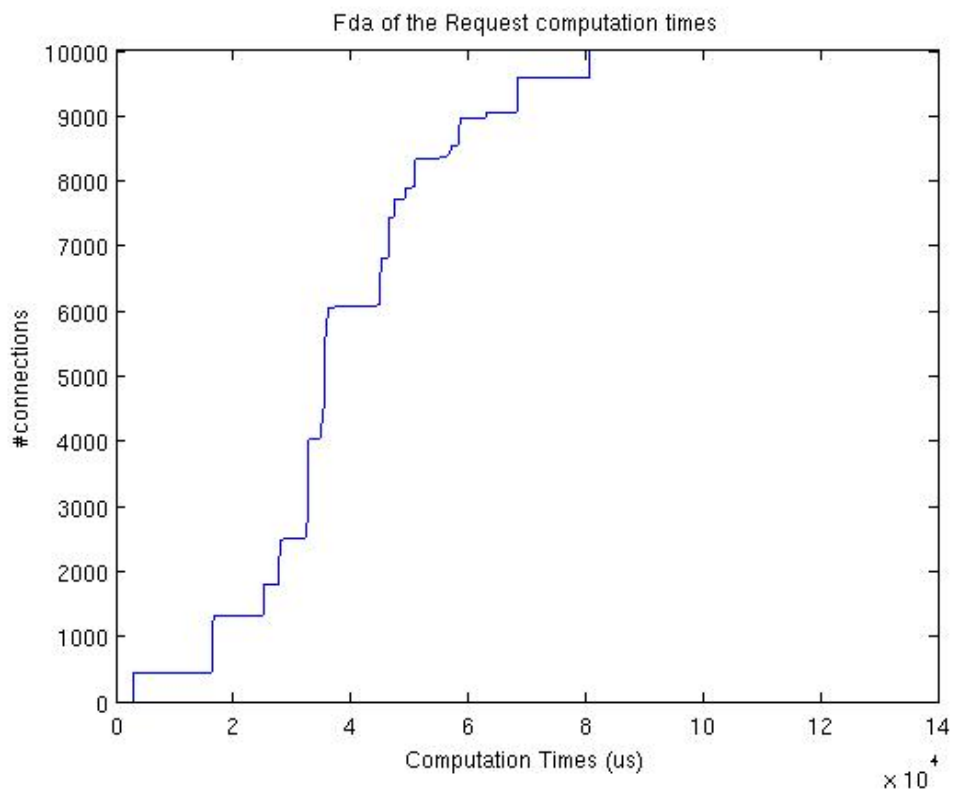


Figura 4-5: Función de distribución acumulada para los tiempos de computación en el CTTC

En las siguientes gráficas se muestran los resultados obtenidos con los tiempos de establecimiento de los LSPs (*Setup Delay Times*) de los caminos computados por el PCE. En la Figura 4-6 y la Figura 4-7) se muestran los resultados del histograma y la función de distribución acumulada de los tiempos resultantes. El tiempo de *Setup* consiste en el tiempo que tarda el plano de control en establecer el LSP requerido mediante el mecanismo del protocolo RSVP. Como referencia vamos a fijarnos en la Figura 3-3 del capítulo anterior. Como podemos ver en esa figura después de la respuesta del PCE con el ERO, en el plano de control se comienza el proceso de comprobación y reserva de recursos nodo a nodo según la secuencia de nodos en el LSP a establecer. El primer nodo comienza este proceso con el envío de un mensaje RSVP Path y a continuación se queda esperando a recibir el mensaje de confirmación RSVP que finaliza el proceso y lo da por válido. Si llega este último mensaje correctamente significa que el LSP ha sido establecido y se puede comenzar con el proceso de actualización de recursos del PCE mediante OSPF.

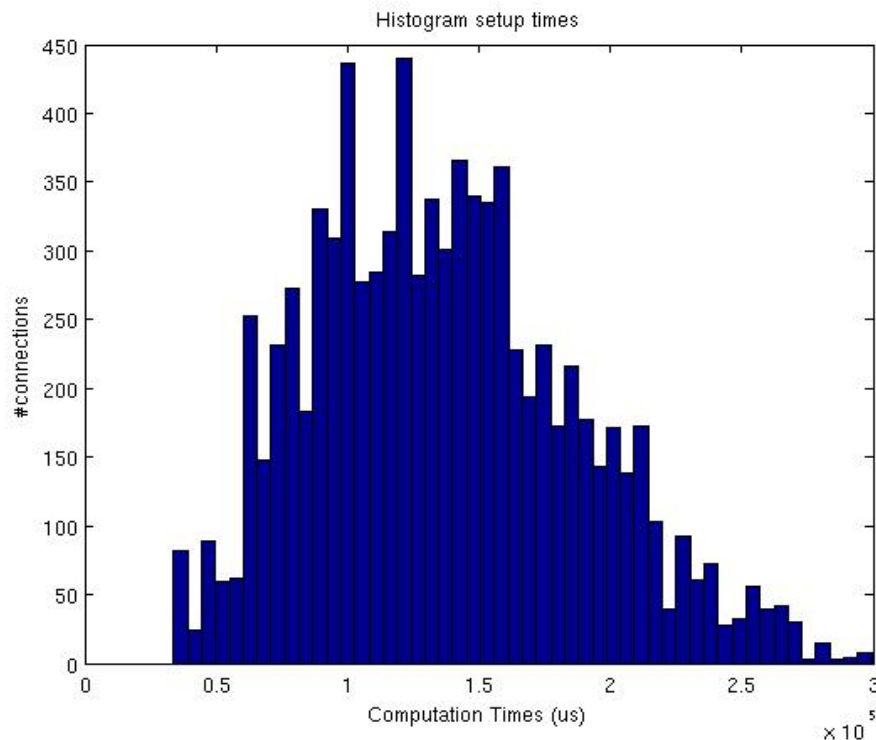


Figura 4-6: Histograma de tiempos de *Setup* de los LSPs en el CTTC

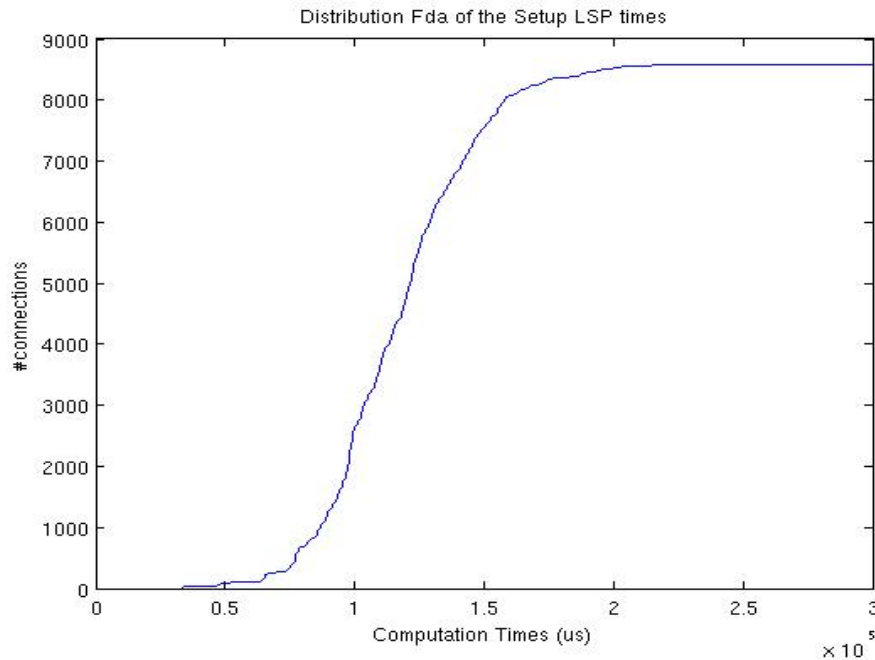


Figura 4-7: Función de distribución acumulada para los tiempos de *Setup* de los LPSs en el CTTC

Como podemos observar estos tiempos siguen una distribución que se acerca a una exponencial y la media de los tiempos de *Setup* la encontramos en 128,710 (ms).

El tiempo de plano de control dependerá en gran medida del número de saltos que tenemos en nuestro camino. En el CTTC tenemos un plano de control “real” distribuido, de tal manera que la reserva de recursos se va haciendo secuencialmente en los elementos de red que componen el camino a establecer. Este tiempo depende por lo tanto de los siguientes aspectos:

1. Comprobación de disponibilidad de recursos en cada elemento de red.
2. Retardos de plano de control para la comunicación entre elementos de red en el proceso del protocolo RSVP. Por lo tanto depende directamente del número de saltos entre nodos que tenemos en el camino a establecer.
3. Tiempos de reserva de recursos en la TED de cada elemento de red.

4.4.2 Resultados TID

Para obtener los resultados en el *testbed* de TID se han empleado simulaciones con tiempo de conexión exponencial de media 3 segundos variando el tiempo entre llegadas para obtener los resultados de carga de red obtenidos. A mayor carga se ha utilizado menor número de muestras, ya que para obtener un valor en los intervalos de confianza al 95% tenemos que tener al menos un número de muestras que sea dos órdenes de magnitud superior a la inversa de la probabilidad de bloqueo obtenida para esa carga. Para los casos sin pre-reserva se han utilizado 300.000 conexiones y para los casos con pre-reserva 150.000. En todas las simulaciones se ha utilizado la topología de la Figura 3-2 con las mismas características físicas especificadas para el *testbed* del CTTC.

Para este *testbed* se han realizado simulaciones con los tres algoritmos (AURE, KSP y KSP-Precomp). Mediante las simulaciones realizadas con el AURE, aplicándose las 5 políticas de selección de λ , se consigue validar lo que figura en el estado del arte [31] sobre estas políticas, tanto para el caso con Pre-Reserva como sin Pre-Reserva. Posteriormente se muestra una comparación entre algoritmos para la misma política de selección de longitud de onda (que en este caso es FF). Mediante esta comparación conseguimos demostrar cómo afectan los tiempos de computación al bloqueo de paquetes cuando aumentan los tiempos de computación. A más tiempo de computación mayor bloqueo de paquetes por “robo de λ ” en el plano de control. Esta comparación se realiza usando los algoritmos KSP y KSP-Precomp con la política de selección de λ FF y sin PreReserva.

Posteriormente en el caso con PreReserva se demuestra como el efecto del “robo de λ ” desaparece y los tiempos de computación no afectan tanto al rendimiento de los algoritmos.

4.4.2.1 Sin Pre-reserva

4.4.2.1.1 Probabilidad de Bloqueo

En primer caso hemos utilizado el AURE para probar cuatro políticas de selección de λ que se proponen en el estado del arte. Como podemos comprobar en la Figura 4-8 se vuelven a cumplir los pronósticos que se obtienen en [31] y se han validado también en el testbed del CTTC. El “exhaustive” nos proporciona el mejor rendimiento de los recursos de la red. Es complicado hacer una comparación “real” con los resultados obtenidos en el CTTC, ya que no se pueden emular el retardo del plano de control del CTCC. Sin embargo, a nivel de algoritmos se comprueba que se llega a los mismos resultados.

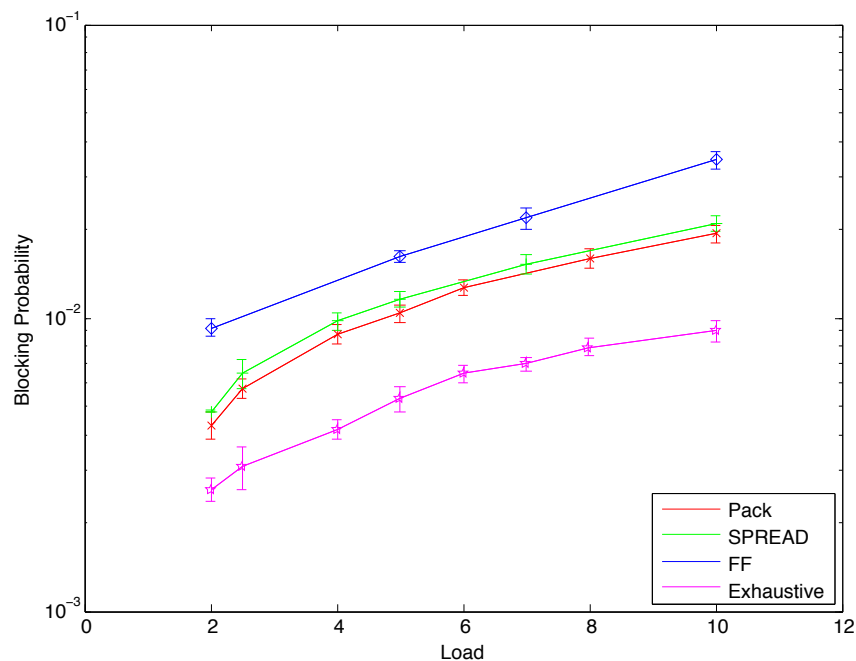


Figura 4-8: Probabilidades de bloqueo AURE para distintas políticas de selección de λ

En la Figura 4-9 se muestra la comparación de los algoritmos AURE, KSP y KSP-Precomp con política de selección de λ fija (FF). Vemos como el caso con pre cómputo de

rutas aporta mejores resultados. Esta diferencia es debida al aumento de bloqueo por “robo de lambda” en el caso sin pre cálculos de caminos, ya que los tiempos de computación son mayores como veremos más adelante. Como es lógico cuanto más tiempo tarda el algoritmo en computar su camino, más probable es que haya un “robo de lambda”, ya que al PCE se le pueden acumular más caminos simultáneos y asignar la misma lambda en un mismo enlace a dos caminos sin que haya tiempo de que llegue el mensaje de actualización del estado del link. Por otro lado podemos ver cómo el algoritmo que más sufre el efecto negativo del “robo de lambda” es el AURE, y por lo tanto ofrece un peor rendimiento en los casos sin pre-reserva a pesar de ser el algoritmo más rápido de los tres.

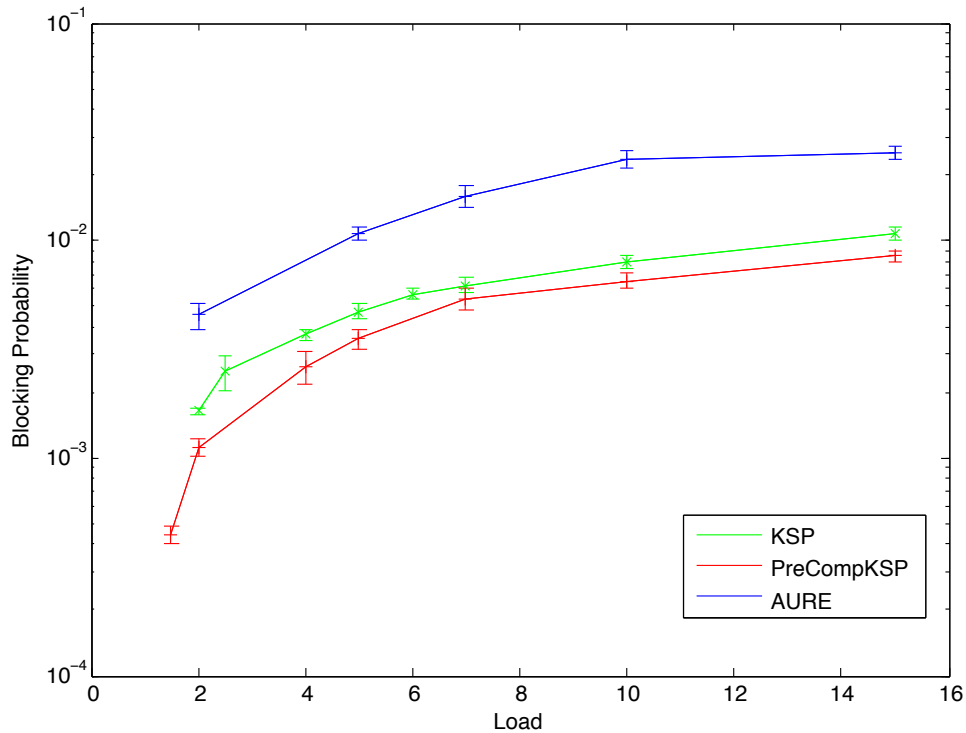


Figura 4-9: Comparación de KSP con pre cálculo, KSP *on live* y AURE para una selección de lambda FF

4.4.2.1.2 Tiempos de Ejecución

Como podemos observar en la Figura 4-10, el tiempo de computación es considerablemente mayor para el caso de KSP “*on live*”. En los casos sin pre-reserva todo el bloqueo se produce por robo de longitud de onda. Por este motivo el tiempo de computación resulta crucial para los resultados que obtenemos, ya que a más tiempo de computación mayor es la probabilidad de que se produzca un robo de lambda. En teoría (y se ha comprobado que de manera práctica también) el KSP y el KSP con pre-cálculo devuelven los mismos caminos sobre nuestra topología, pero la diferencia es que el pre-cálculo nos reduce los tiempos posteriores de computación en directo, por lo que conseguimos reducir el robo de longitud de onda y, de la misma manera, el bloqueo en la red. En la Figura 4-10 están presentes los intervalos de confianza pero son tan pequeños que no son apreciables.

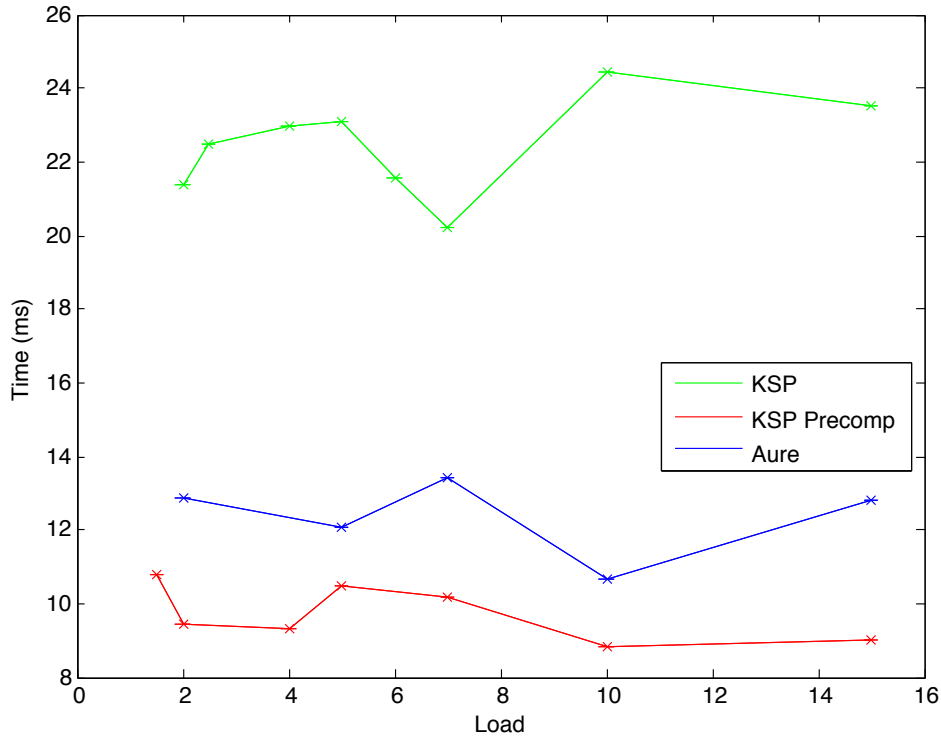


Figura 4-10: Tiempos de computación sin Pre Reserva

Si nos fijamos a nivel computacional en cada algoritmo podemos obtener la solución sobre la diferencia de tiempos de computación que obtenemos entre el KSP y el AURE. Una vez tenemos la ruta seleccionada y hemos hecho la selección de posibles lambdas ordenadas según el criterio aplicado, en el caso del KSP tenemos que ir comprobando si la lambda resultante tiene continuidad física en todo el camino computado. Para ello tenemos que ir revisando la ocupación de dicha lambda en cada *link* de la ruta. Esta operación es la que hace que computacionalmente los tiempos aumenten, y por ello nos encontramos con que el AURE funciona de manera más rápida que el KSP. Detallar que los intervalos de confianza en este caso son tan insignificantes que no se pueden llegar a apreciar en las gráficas de tiempos.

4.4.2.2 Con Pre-reserva

4.4.2.2.1 Probabilidad de Bloqueo

Mediante la mecanismo de pre-reserva, el bloqueo por “robo de longitud de onda” (ver sección 4-3) se elimina por completo. Esto nos permite aumentar muy considerablemente la carga de la red hasta que encontramos bloqueo por no existir camino en la red. De nuevo volvemos a ver en la Figura 4-11 cómo las políticas de selección de lambda funcionan de la misma manera que en el estado del arte [31], produciendo los mismos beneficios las unas con las otras.

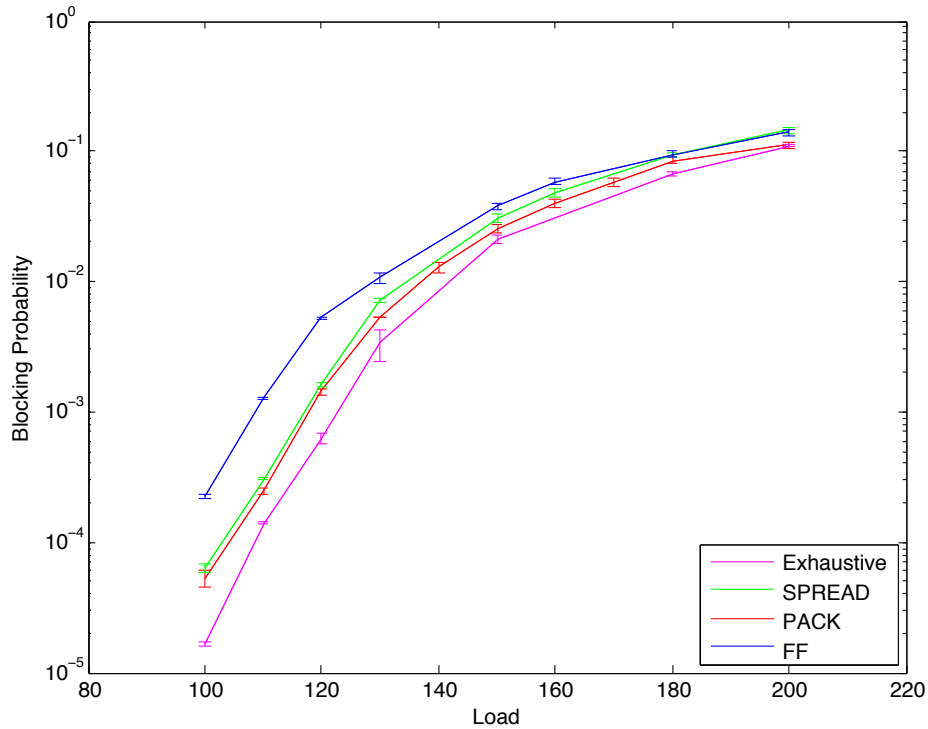


Figura 4-11: Probabilidad de bloqueo del AURE con distintas políticas de selección de lambda con Pre Reserva

Cuando realizamos la comparación entre el KSP en vivo y el KSP con pre-cómputo utilizando la pre-reserva, no obtenemos ninguna mejora con el pre cómputo (Figura 4-12), ya que al estar eliminando el robo de lambda mediante la reserva, el tiempo de computación no afecta a la probabilidad de bloqueo. Como podemos ver el AURE con pre-reserva es el algoritmo que aporta los mejores resultados de rendimiento para la red, esto es debido a que optimiza mucho más los recursos de la misma realizando una computación más extensa.

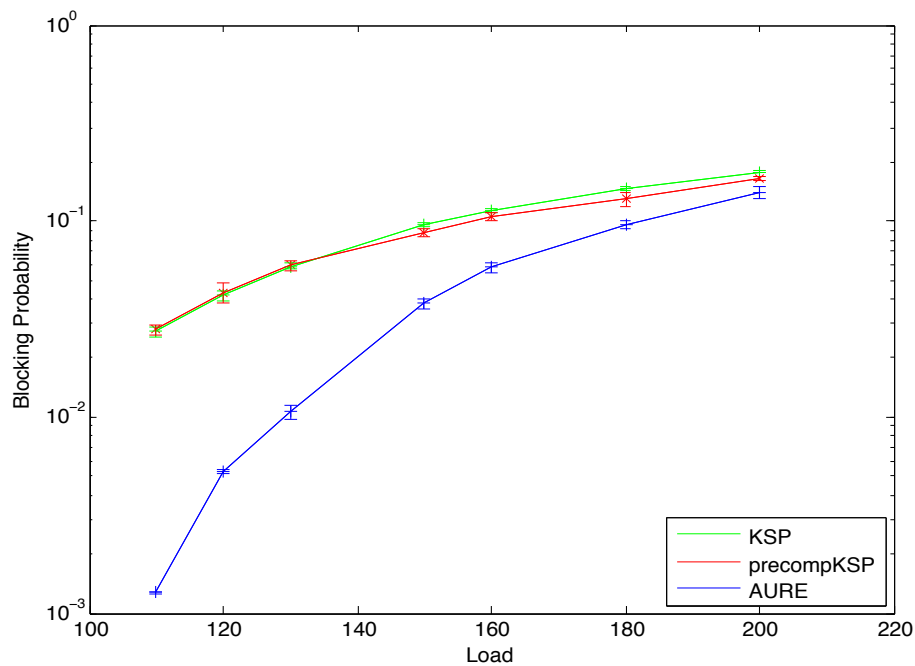


Figura 4-12: Comparación Probabilidad de bloqueo KSP *on live*, KSP *Precomp* y AURE con Pre Reserva y selección de lambda FF

4.4.2.2 Tiempos de ejecución

Como podemos ver en la Figura 4-13, el mecanismo de pre-reserva ralentiza notoriamente los tiempos de computación de caminos en el PCE. Por ello podemos ver como comparándolos con los tiempos sin pre-reserva, éstos han aumentado en torno a 30 ms para los tres algoritmos. Esto es debido a que al activar el mecanismo de pre-reserva, después de cada computación tenemos que modificar la TED activando los bits del *bitmap* de cada enlace necesarios para activar la reserva de recursos. De nuevo los intervalos de confianza son tan pequeños que son prácticamente inapreciables en la Figura 4-13. Estos intervalos de confianza son para el 95% de las muestras totales.

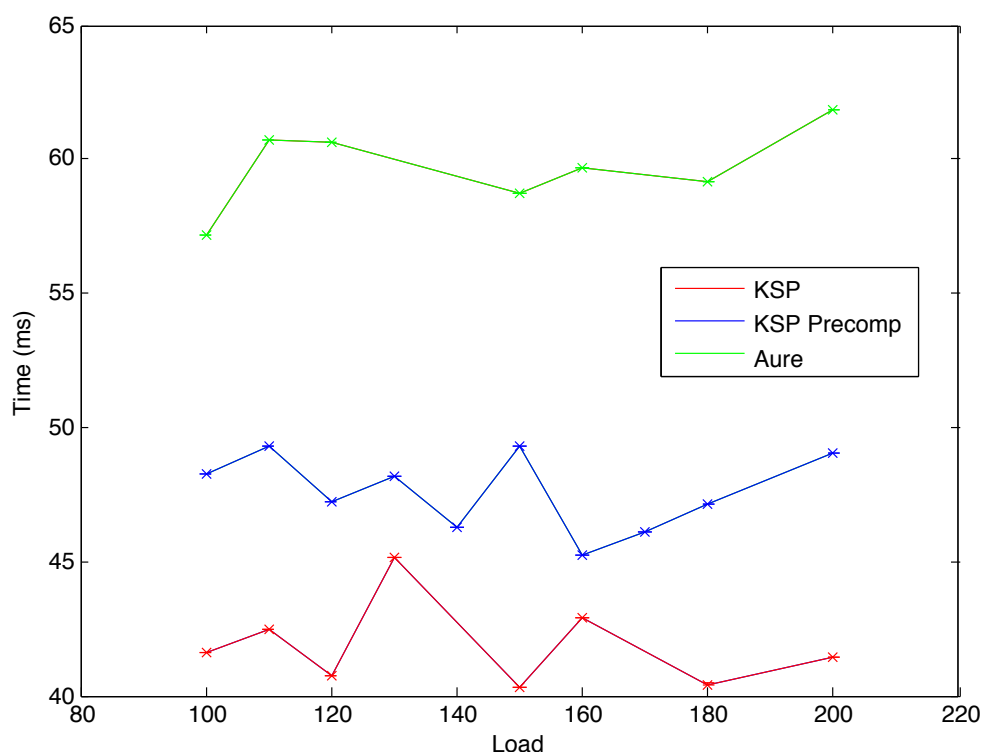


Figura 4-13: Tiempos de computación con Pre Reserva

5 Evaluación de algoritmos para redes multicapa

A la hora de elaborar algoritmos multicapa nos encontramos con una serie de problemas que tenemos que afrontar. Estos problemas a solucionar son los siguientes:

1. Determinar la topología virtual formada por los *light-paths*.
2. Encaminar esos *light-paths* sobre la topología física. De la misma manera como hacíamos con los algoritmos de WSON, definimos *light-paths* como TE-Links creados en la capa IP/MPLS. Una vez creados pueden ser utilizados como links de datos normales sobre los que encaminar el tráfico.
3. Aplicar políticas de asignación de longitud de onda para los *light-paths* ya encaminados.
4. Encaminar el tráfico que recibe la red sobre la topología virtual.

Una vez mostrados estos cuatro sub-problemas tenemos dos soluciones: tratar de resolverlos en orden y de manera independiente o tratarlos como un todo. En caso de tratarlos como problemas separados, tenemos que tener un modelo de capas separadas para el PCE Multicapa (modelo separado). Es decir, cada capa como un grafo separado, con una correspondiente TED separada. Esta solución es menos eficiente, ya que no se puede combinar de manera óptima la información sobre de la capa con la topología física y la capa con la topología virtual para determinar el encaminamiento del tráfico. Sin embargo, esta solución es más escalable ya que reduce la complejidad del problema y el tiempo de cálculo es menor al tener grafos más pequeños.

La segunda solución es ver la red como un modelo de grafo único (modelo integrado) a la hora de computar los caminos. Para ello, como detallaremos más detenidamente en este apartado, tenemos que construirnos un grafo auxiliar completo de toda la red multicapa. De esta forma podemos tener información sobre los cuatro sub-problemas de manera integrada. Este modelo tiene mayor potencial para obtener unos mejores resultados.

5.1 Modelo Separado

En una red de entorno multicapa, para una petición de conexión dada en un entorno no integrado, tenemos que plantearnos las siguientes cuestiones para poder llegar a la solución más óptima:

- Tenemos que ver primero si esta conexión puede ser establecida sobre algún link de la actual topología virtual. En este punto, se puede elegir establecer la demanda siguiente sobre la topología virtual o crear un nuevo *light-path*. El problema es que tal vez pueda ser más eficiente establecer un nuevo *light-path*. Nos encontramos aquí con un problema de optimización de la red y de los recursos de ésta.
- En caso de que no tengamos conexiones directas entre los nodos IP de origen y destino sobre la topología virtual, tenemos que estudiar cómo modificar o crear nuevos *light-paths* en esta capa para poder acomodar eficientemente la conexión. Tal vez sea mejor, como se ha explicado en el caso anterior, crear una conexión directa entre nodos a través de la capa óptica, o puede que sea mejor establecer

conexiones entre nodos intermedios y utilizar los *light-paths* ya creados previamente para establecer un camino óptimo conjunto.

Las diferentes decisiones que tomemos sobre estas cuestiones serán las que nos darán como resultado distintos rendimientos sobre nuestra red multicapa. A partir de ahora nos vamos a referir a estas políticas de encaminamiento como *Grooming Policies*, y son las que nos dan lugar a los distintos algoritmos que presentamos aquí en esta sección.

A la hora de aplicar estas políticas (*Grooming Policies*) disponemos de cuatro operaciones de encaminamiento [32] básicas, cuyas características resumidas podemos observarlas en la Tabla 5.1.

	Añadir nuevo(s) <i>light-path(s)</i>	Política de encaminamiento con salto único o multi-salto
Operación 1	No	Salto único
Operación 2	No	Multi-salto
Operación 3	Sí	Salto único
Operación 4	Sí	Multi-salto

Tabla 5-1: Tabla de comparación de las cuatro operaciones

A continuación detallamos la utilidad de cada operación a la hora de encontrar un camino para encaminar el tráfico:

- **Operación 1.** Encamina el tráfico sobre un *light-path* ya existente que conecte directamente los nodos origen y destino.
- **Operación 2.** Ruta el tráfico sobre múltiplex *light-paths* ya creados previamente.
- **Operación 3.** Establece un nuevo *light-path* directo entre los nodos origen y destino y ruta el tráfico a través de este camino. Con esta operación establecemos un solo *light-path* en caso de que la cantidad de tráfico sea menor que la capacidad del link.
- **Operación 4.** Establece uno o más *light-paths* que no conecten directamente los nodos origen y destino, y ruta el tráfico sobre estos *light-paths* y/o sobre algunos ya creados previamente. Al usar esta operación tenemos que crear como mínimo un *light-path*.

En algunas ocasiones todas las operaciones son posibles, por lo que tenemos que aplicar una *Grooming Policy* para llevar a cabo la selección de caminos. Los algoritmos implementados en este caso son los siguientes:

- **Minimizar el número de saltos de tráfico (MinTH).** Esta política minimiza en número de saltos que hay en la red. Para ello, empezamos probando la operación 1. Si ésta falla siempre intentamos establecer un nuevo *light-path* directo entre los nodos origen y destino. Solamente en caso de que no pueda ser establecido utilizamos una política de multi-salto. Esta política se consigue mediante la aplicación de la operación 1, seguida de la operación 3 en caso de que ésta falle. Si ambas fallan habría que llamar a las operaciones 2 y 4 y elegir el camino con menor número de saltos en la topología virtual.

- **Minimizar el número de *Light-paths* (MinLP).** Esta política tiene como objetivo minimizar la creación de nuevos *light-paths* para encaminar el tráfico entrante. En este caso si la operación 1 falla, intentamos encaminar la conexión utilizando múltiples *light-paths* ya existentes (operación 2). Si la operación 2 también falla, primero intentamos encaminar utilizando la operación 3 y, en caso de que no haya camino, utilizamos la operación 4.
- **Minimizar el número de *Wavelength-Links* (MinWL).** Esta política intenta consumir el mínimo número de *Wavelength-Links* (WL) para encaminar el tráfico. En este caso si las operaciones 1 y 2 fallan, el algoritmo compara el número de WLs usados en las operaciones 3 y 4, y elige el camino que requiera menor número de WLs.

Si ninguna operación puede ser realizada es cuando se da el caso de bloqueo de tráfico. Llegados a este caso podríamos aplicar un reordenamiento de los *light-paths* para encontrar un camino al tráfico bloqueado, pero esto causaría el estancamiento de todo el tráfico entrante durante esta operación de reconfiguración de caminos. En este proyecto no vamos a tener en cuenta esta política, por lo que no aplicamos reconfiguración de *light-paths*.

5.2 Modelo Integrado

Cuando se tiene un modelo integrado se utiliza un grafo auxiliar completo con todas las conexiones en la red. A este grafo le asignamos una serie de pesos para posteriormente poder computar el algoritmo *K-Shortest-Path* en el mismo, ajustándonos a los criterios de cada *Grooming policy* definida. Mediante la asignación de pesos a cada enlace de este grafo conseguimos reducir la utilización de los enlaces requeridos en la aplicación de cada política de encaminamiento.

En el test-bed de TID, esta separación de capas es solo efectiva de cara al PCE Multicapa. El VNTM y el NMS siguen gestionando cada uno su capa con su grafo correspondiente.

5.2.1 Construcción del Grafo Auxiliar

Para mostrar la construcción del grafo auxiliar nos vamos a servir de un ejemplo con una topología muy simple (Figura 5.1) [31]. Tenemos una red de 3 nodos con 4 links unidireccionales. Al comenzar no tenemos ningún *light-path* en la red, por lo tanto la topología virtual está vacía inicialmente (Figura 5.2). Cada link posee dos longitudes de onda.

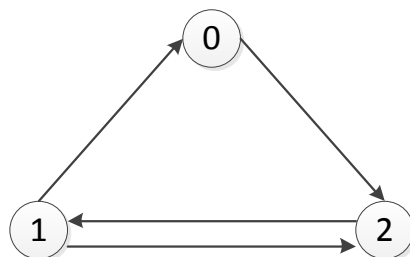


Figura 5-1: Topología física (óptica) de la red de ejemplo

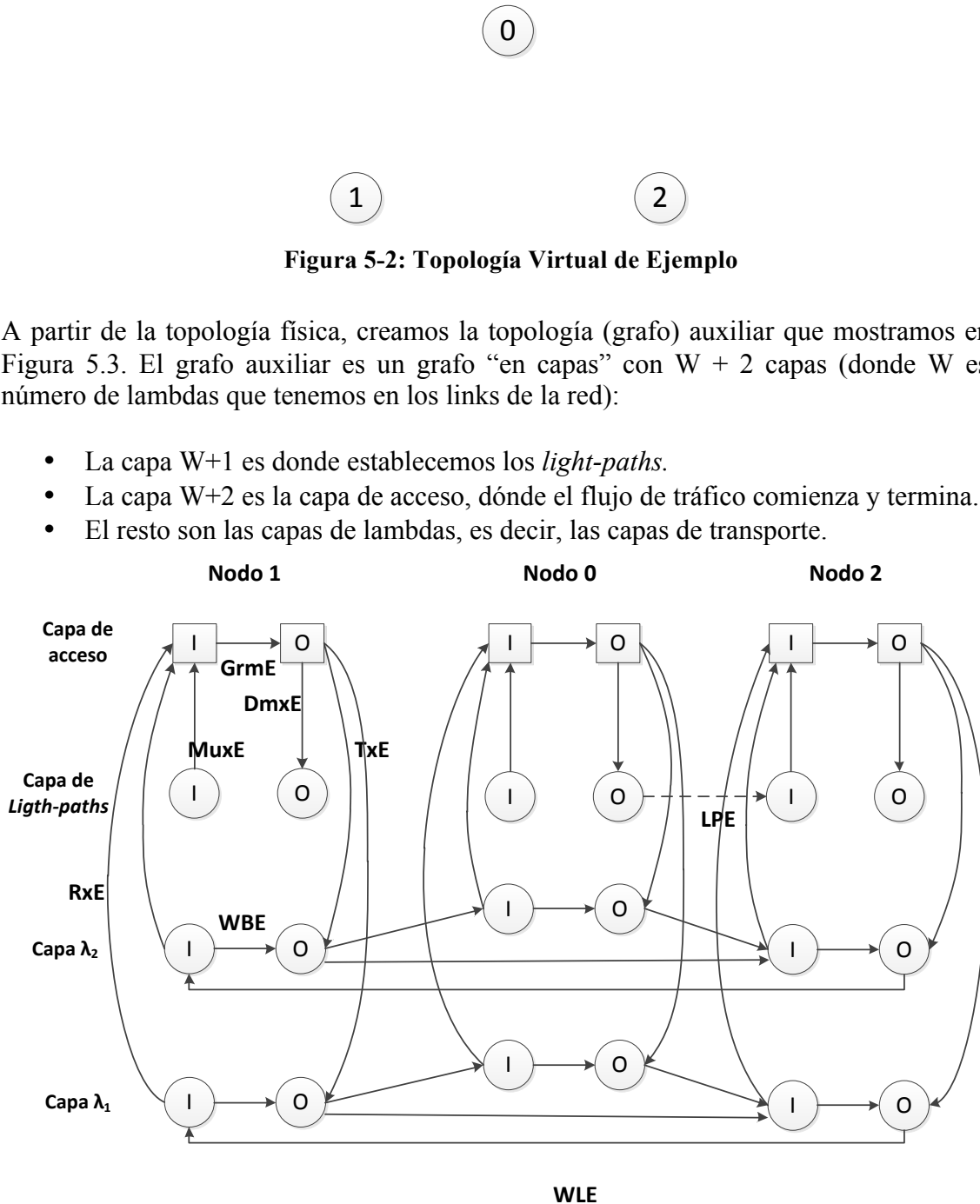


Figura 5-3: Grafo Auxiliar de la red de ejemplo

Cada nodo tiene dos puertos en cada capa, uno de entrada u otro de salida, y cada link en el grafo auxiliar tiene dos campos asociados: la capacidad del link y el peso. Para facilitar la utilización del grafo auxiliar le damos a cada puerto de entrada/salida (en todo el grafo completo) una dirección IPv4 diferente con el siguiente formato:

```
baseIP = "172.20."+layer+".";
```

Figura 5-4: Dirección IPv4 de los nodos del Grafo Auxiliar

Como podemos ver los dos primeros campos de la dirección son idénticos. El tercer campo de la dirección lo ajustamos al número de capa en el que se encuentre el puerto, y el cuarto a la posición del puerto (nodo del grafo auxiliar) dentro de la capa en la que se encuentre, empezando por el cero. Por otro lado tenemos también un identificador para cada nodo del grafo auxiliar, empezando por el cero también hasta el número máximo de nodos que tengamos en el grafo auxiliar.

Es necesario guardar las relaciones entre nodos “reales” y los nodos del grafo auxiliar para posteriormente poder hacer las conversiones entre EROs de la manera más simple. Por ello hay que ir creando estructuras de propiedades para obtener rápidamente la información esencial de cada nodo del grafo auxiliar:

```
public class InfoNodo{ //Nodes --> (layer, I/0, nodeIP)
    private int numLayer;
    private int I_0;    // 1 -->
    private Inet4Address IPnode;

    public int getnumLayer(){
        return this.numLayer;
    }
    public void setnumLayer(int numLayer){
        this.numLayer = numLayer;
    }
    public int getI_0(){
        return this.I_0;
    }
    public void setI_0(int I_0){
        this.I_0 = I_0;
    }
    public Inet4Address getIPnode(){
        return this.IPnode;
    }
    public void setIPnode(Inet4Address IPnode){
        this.IPnode = IPnode;
    }
}
```

Figura 5-5: Propiedades asociadas a cada nodo

Y posteriormente asociar a cada identificador de cada nodo su tabla con la información correspondiente:

```
//table of relation of nodes and info in the AuxGraph
public Hashtable<Integer, InfoNodo> infoTable;
```

Figura 5-6: *Info Table* para la relación entre los nodos auxiliares y sus propiedades

También guardamos las relaciones entre las direcciones IPv4 de los nodos auxiliares y los identificadores de esos nodos en tablas asociadas:

```
public Hashtable<Integer, Inet4Address> info_IntNodes_IPv4;
public Hashtable<Inet4Address, Integer> info_IPv4_IntNodes;
```

Figura 5-7: Relaciones entre los nodos del grafo auxiliar y sus identificadores

Con estas relaciones podemos sacar fácilmente la lambda utilizada en la topología real y los nodos reales de la red multicapa a partir del ERO obtenido con la computación sobre la topología del grafo auxiliar. Una vez tenemos los nodos “reales” podemos construir el ERO “real” añadiendo los sub-objetos necesarios según el tipo de ERO obtenido.

A continuación se detallan los tipos de enlaces que tenemos en el grafo auxiliar, tal y como se definen en [31]:

- *Wavelength Bypass Edges* (WBE): Son los enlaces que van de cada puerto de entrada a cada puerto de salida de cada nodo en las capas de lambdas (capas de transporte).
- *Grooming Edges* (GrME): Los enlaces que van desde los puertos de entrada a los de salida en los nodos de la capa de acceso.
- *Mux Edges* (MuxE): Los enlaces que van desde el puerto de salida del nodo en la capa de acceso, al puerto de salida del nodo en la capa de *light-paths*.
- *Transmitter Edges* (TxE): Los enlaces que van desde los puertos de salida del nodo en la capa de acceso, al puerto de salida de los nodos en las capas de lambdas.
- *Receiver Edges* (RxE): Los enlaces que van desde los puertos de entrada en las capas de lambdas a los puertos de entrada en los nodos de la capa de acceso.
- *Wavelength-Link Edges* (WLE): Los enlaces que van desde los puertos de salida en los nodos de las capas de lambdas, a los puertos de entrada de los nodos en las capas de lambdas (en caso de que haya un enlace físico entre ambos nodos).
- *Light-path Edges* (LPE): Los enlaces que van desde el puerto de salida de un nodo en la capa de *light-paths* al puerto de entrada de otro nodo en la misma capa, en caso de que el *light-path* haya sido establecido previamente.

Como paso final a la construcción del grafo auxiliar, tenemos que asignar los pesos a cada enlace. Estos pesos representan el coste de encaminar una conexión por cada elemento de la red, y de cada política de encaminamiento a nivel global.

5.2.2 Asignación de pesos al Grafo Auxiliar

Para asignar unos valores correctos que hagan corresponder cada simulación a lo que se tiene que ajustar según la política de encaminamiento deseada, tenemos que hacer un estudio sobre qué enlaces se usarían mayoritariamente en cada operación usada. De esta manera podemos ver posteriormente el uso de qué enlaces hay que minimizar al aplicar cada política de encaminamiento. Mostramos los resultados a continuación para cada operación seleccionada [32]:

- La **Operación 1** utiliza un único LPE para rutar el tráfico. Esta operación encamina el tráfico en enlaces directos creados en la topología virtual. Como cada LPE siempre tiene un enlace MuxE y otro DmxE conectados el peso de esta operación será:

$$MuxE + LPE + DmxE.$$

- La **Operación 2** usa n ($n \geq 2$) LPEs para rutar el tráfico. Esta operación encamina el tráfico en los enlaces creados en la topología virtual, que no sean directos. De esta manera el peso del camino con esta operación será:

$$n \times (MuxE + LPE + DmxE) + (n - 1) \times GrmE.$$

- La **Operación 3** establece un nuevo *light-path* entre el origen y el destino. Este *light-path* se basa en un camino encontrado en las capas de lambdas del grafo, que consiste de los siguientes enlaces: un TxE, m ($m \geq 1$) WLEs y un RxE. Por lo tanto el peso será el siguiente:

$$TxE + m \times WLE + (m - 1) \times WBE + RxE.$$

- La **Operación 4** mezcla la creación de nuevos *light-paths* (k , donde $k \geq 1$) con otros enlaces en la topología virtual ya existentes (k' , donde $k' \geq 0$). Suponiendo que cada LPE nuevo usa l_i ($l_i \geq 1$, $1 \leq i \leq k$) WLEs, el peso del camino es el siguiente:

$$\begin{aligned} & \sum_{i=1}^k (TxE + l_i \times WLE + (l_i - 1) \times WBE + RxE) \\ & + k' \times (MuxE + LPE + DmxE) \\ & + (k + k' - 1) \times GrmE \end{aligned}$$

Basándonos en el análisis de tipos de enlaces asociados a cada operación podemos manipular los pesos de cada enlace para satisfacer las distintas políticas de encaminamiento. Ello se consigue dando un peso alto a los enlaces cuyo uso hay que minimizar en el uso de cada operación. A continuación se muestran las tres políticas con los enlaces que debemos usar menos en la computación:

- **MinTH.** Esta política minimizar el uso de enlaces GrmE. Por lo tanto para aplicar esta política tenemos que dar un valor alto al peso de este tipo de enlaces.
- **MinLP.** Al aplicar esta política se minimiza el uso de los enlaces del tipo TxE y RxE. Para ello también les damos un valor alto a su peso.
- **MinWL.** Se minimiza el uso de los enlaces de tipo WLE

Tenemos que tener en cuenta que para todos los algoritmos la Operación 1 es siempre la primera opción tenemos que el valor asignado al peso de esta operación es siempre el menor para los tres algoritmos. Por ello empezamos fijando a los LPE un peso de 10 unidades. Basándonos en los resultados obtenidos previamente, asignamos los pesos a los enlaces restantes en la Tabla 5-2 [31].

	MinTH	MinLP	MinWL
WLE	10	10	1000
GrmE	1000	20	0
TxE	20	200	20
RxE	20	200	20
LPE	1	1	1
MuxE	0	0	0
DmxE	0	0	0
WBE	0	0	0

Tabla 5-2: Tabla de pesos asociados a cada link en los experimentos para las tres políticas de encaminamiento

5.3 Implementación y procedimientos de los algoritmos expuestos para cada modelo

Como hemos dicho previamente, los algoritmos se obtienen de la combinación de las diferentes operaciones de las que hemos explicado previamente (dos TEDs separadas) y de la asignación de pesos a los distintos enlaces del grafo auxiliar (una TED integrada). Esta sección explica como se implementan las políticas, primero en el modelo separado y luego para el integrado.

5.3.1 Modelo separado

En el caso de tener dos PCEs, uno para capa, hay que hacer uso de las operaciones explicadas previamente aplicándolas a cada capa según corresponda. Para ello se siguen los siguientes procedimientos para cada una de ellas:

- **Operación 1.** Lo primero que tenemos que hacer es duplicarnos el grafo de la capa IP/MPLS y podar los links que tengan menor capacidad que el ancho de banda que estamos solicitando en la petición. Una vez hecho esto simplemente tenemos que correr el algoritmo *KSP* sobre el grafo podado de la topología. Si no encontramos camino o el camino tiene más de un salto pasamos a otra operación.
- **Operación 2.** Realizamos la misma operación de duplicado y podado con el grafo de la capa IP/MPLS de la Operación 1. En este caso solo pasamos a otra operación si no existe camino o solo hay caminos de un salto, independientemente del número de saltos que tenga el camino calculado.
- **Operación 3.** En esta operación primeramente tenemos que obtener los nodos correspondientes, en el grafo de la capa óptica, a los nodos origen y destino de la capa IP/MPLS con los que me están realizando la petición de camino. Esta correspondencia la obtenemos a partir del grafo *Inter-layer*. Una vez obtenidos estos nodos tenemos que correr el algoritmo AURE de WSON sobre el grafo de la topología óptica
- **Operación 4.** Esta es la operación más complicada, ya que tenemos que optimizar el camino utilizando links y nodos de ambas capas, y por lo tanto de los dos grafos (tanto el de la topología IP/MPLS como el de la Óptica). En caso de llegar al uso de esa operación es porque no tenemos ningún camino físico directo sobre la capa IP/MPLS. Para solucionarlo tenemos que sacar los árboles de nodos conectados, tanto por el lado del nodo origen como por el nodo destino como se muestra en la Figura 5-8.

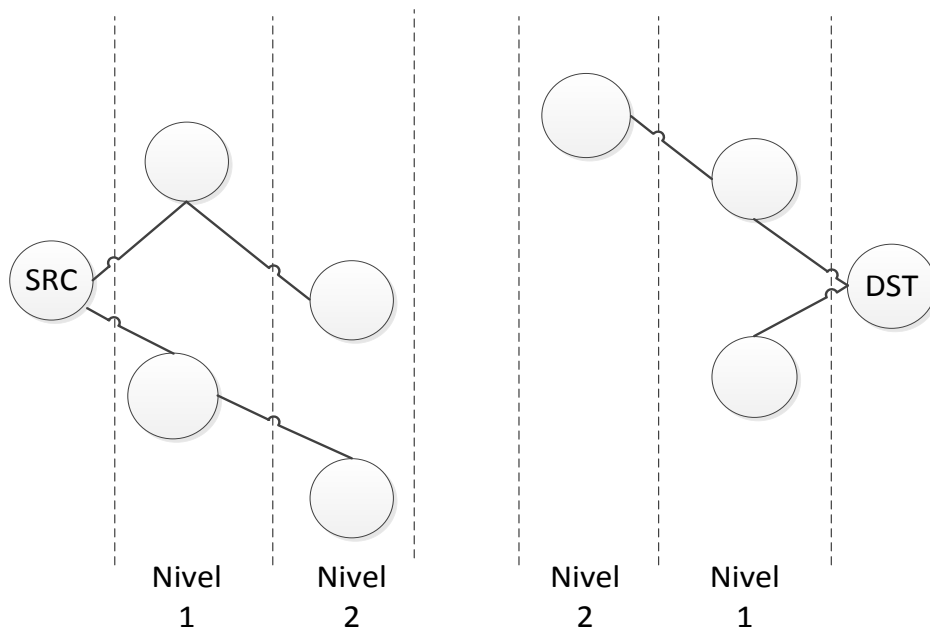


Figura 5-8: Árboles origen y destino en la Operación 4

Una vez tenemos contruidos los dos arboles (origen y destino) del grafo IP/MPLS, tenemos que ir avanzando sobre los nodos de ambos e ir computando el algoritmo AURE sobre el grafo de la capa óptica, realizando previamente la conversión de nodos (para los nodos origen y destino) para obtener los correspondientes nodos ópticos. Es decir tenemos que computar caminos entre todos los pares de nodos disponibles entre un árbol y el otro. Una realizado esto nos quedaremos con el camino que tenga menor número de saltos en la capa IP/MPLS.

Una vez tenemos implementadas las operaciones, simplemente tenemos que hacer uso de ellas según el orden indicado para cada algoritmo (ver sección 5.1) haciendo uso de las dos TEDs que tenemos, una para cada capa.

5.3.2 Modelo integrado

En el caso de un PCE multicapa el proceso es sencillo. Para cumplir con las políticas explicadas anteriormente los autores en [31] asignan los pesos mostrados en la Tabla 5-2. Como la Operación 1 debe ser siempre la primera opción, por lo que el valor asignado al peso de esta operación es siempre el menor para las cuatro operaciones.

Estos pesos se tienen que asignar al grafo auxiliar en el pre cómputo y posteriormente seguir los siguientes pasos:

- Primero se muestran los pasos de computación en el grafo auxiliar mediante el KSP:
 1. Seleccionar una demanda de tráfico entrante con una cierta granularidad (ancho de banda).
 2. Realizamos una copia del grafo auxiliar y lo recorremos para encontrar los enlaces que tienen suficiente ancho de banda. Estos enlaces corresponden a la capa $W+1$, que es la de los *light-paths*. Una vez encontrados borramos los

enlaces cuya capacidad sea menor a la granularidad del ancho de banda de la petición.

3. Buscar el camino más corto desde el puerto de salida del nodo origen en la capa de acceso, hasta el puerto de entrada del nodo destino en la misma capa. Una vez realizado esto el grafo duplicado se borra y para la próxima petición seguimos teniendo el original con todos los *light-paths* originales.

5.4 Evaluación de los resultados

Para la realización de las pruebas vamos a simular con dos escenarios (ver sección del plano de control de TID):

- Topología de Telefónica España.
- NSF Network 14: estos resultados se encuentran en el Anexo A, y se usan para comparar y validar con los de la red de Telefónica España.

5.4.1 Validación de las distintas políticas

Para la validación de resultados se ha utilizado solamente las simulaciones con la topología de Telefónica España, con 80 lambdas por fibra óptica y 2 TEDs se obtiene la validación de las características de los algoritmos. Para ello se introducen en la red 400 conexiones estáticas obteniendo las siguientes gráficas resultados (Figura 5-9 para 2 TEDs y Figura 5-10 para una TED). Se han utilizado 400 peticiones para obtener resultados sin bloqueo y que se permitan ver las estadísticas obtenidas en las políticas de manera correcta sin que tengan que diferir en su funcionamiento por bloqueo de paquetes:

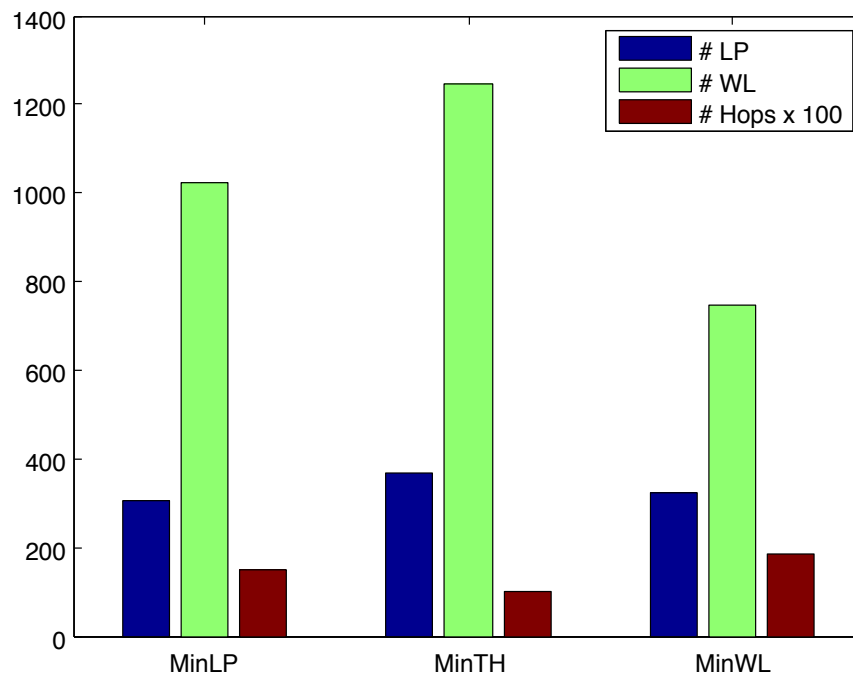


Figura 5-9: Comparación de políticas mostrando utilización de recursos en 2 TEDs

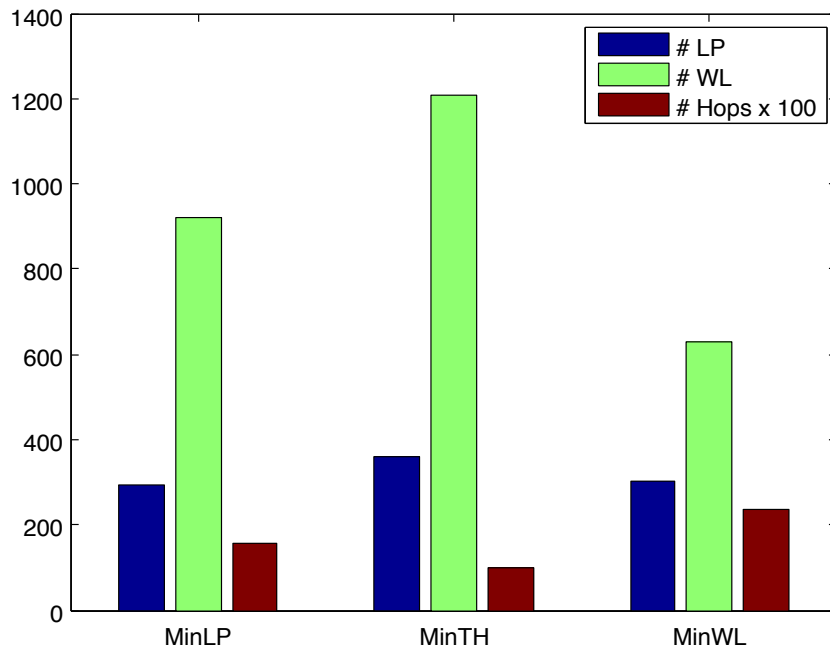


Figura 5-10: Comparación de políticas mostrando utilización de recursos en 1 TED

Como podemos ver cada política usada minimiza los recursos a los que hace alusión en su definición tanto para el caso de separado e integrado. Para obtener estos resultados se ha utilizado un modelo estático sin bloqueo de paquetes en la topología de Telefónica España. También se puede observar cómo, comparando ambos gráficos en el caso de 1 TED se necesitan menos *wavelength-links* (WLs) para establecer las 400 conexiones en la red. Esto da lugar a que, a largo plazo, se puedan establecer en la red mayor número de conexiones sin obtener bloqueo.

Para profundizar más aún en la demostración de los resultados obtenidos en el caso multicapa, se han obtenido estadísticas sobre el número de operaciones que se utiliza en cada modelo y en cada algoritmo de encaminamiento (Figura 5-20 y Figura 5-22), así como el tanto por ciento que supone cada cantidad de operaciones sobre e total usadas (Figura 5-21 y Figura 5-23).

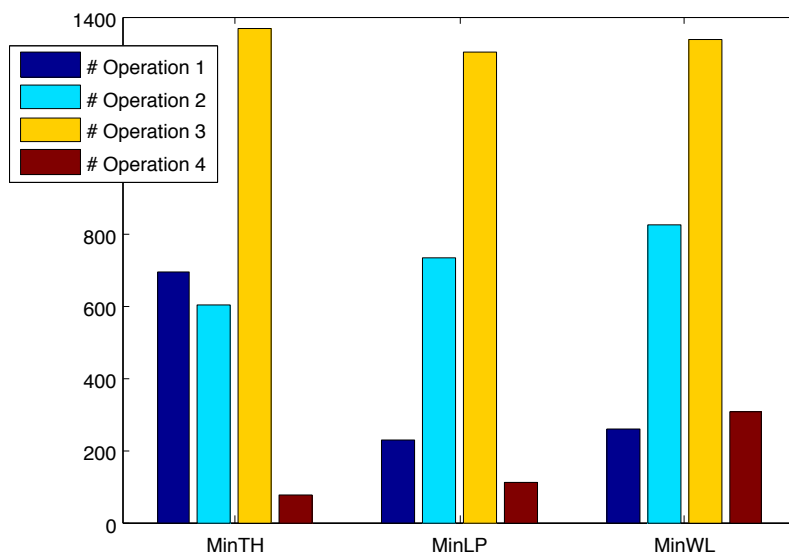


Figura 5-11: Gráfica del número de operaciones en cada algoritmo para el modelo de 1 TED integrada

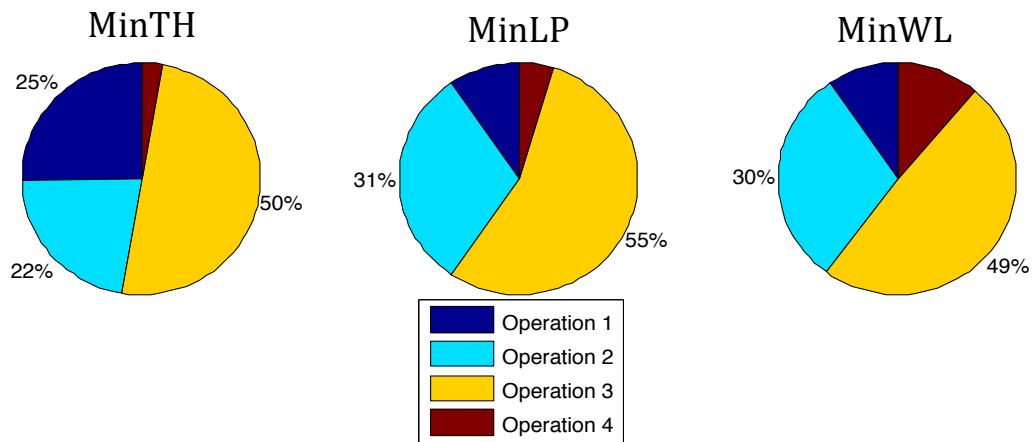


Figura 5-12: Resultados de proporción de operaciones usadas en cada algoritmo para el modelo de 1 TED integrada

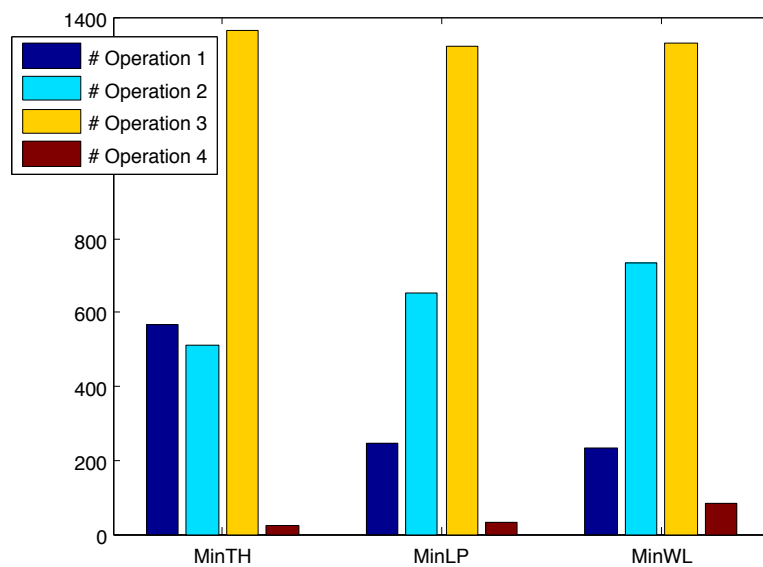


Figura 5-13: Gráfica del número de operaciones en cada algoritmo para el modelo de 2 TEDs separadas

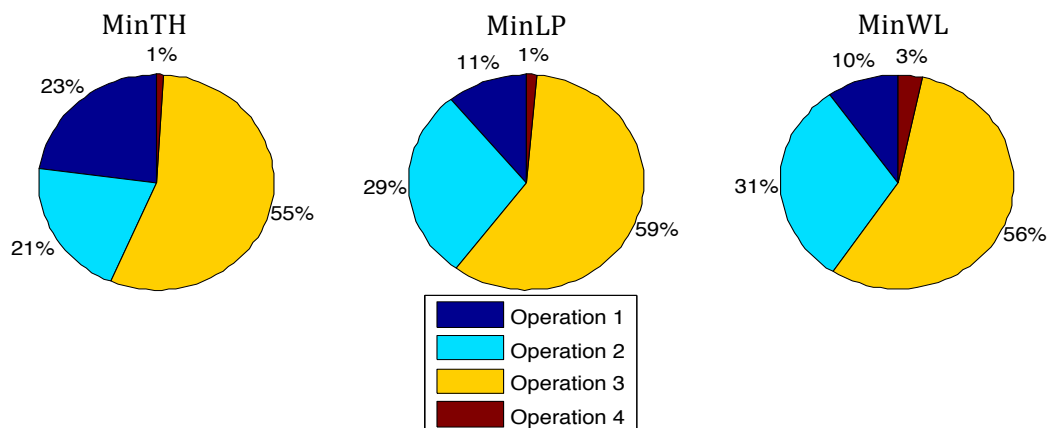


Figura 5-14: Gráfica del número de operaciones en cada algoritmo para el modelo de 2 TED integrada

Podemos ver como cada política maximiza el uso de cada operación, de tal forma que se satisfagan las condiciones que se planteaban para cada uno.

A continuación se muestran (Figura 5-15, Figura 5-16, Figura 5-17 y Figura 5-18) la evolución en el uso de cada una de las operaciones para las tres políticas de encaminamiento y en los dos modelos de arquitectura (TED integrada y 2 TEDs separadas).

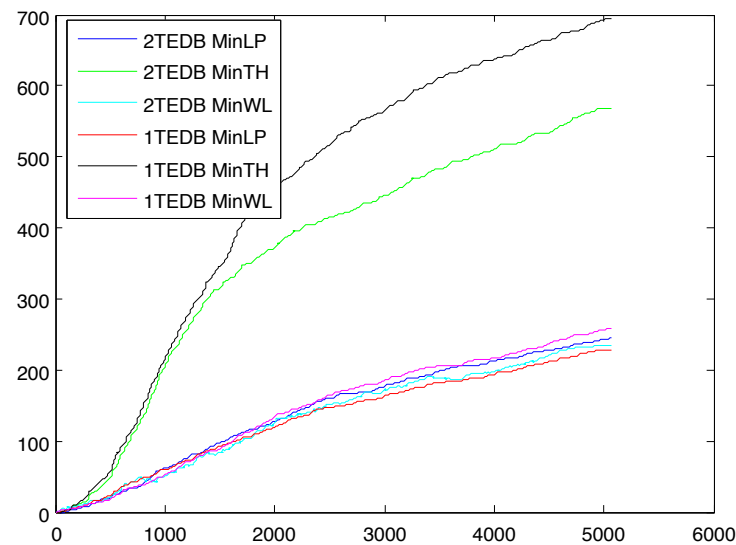


Figura 5-15: Evolución número de operaciones 1 para los distintos casos en simulación estática con 5000 conexiones

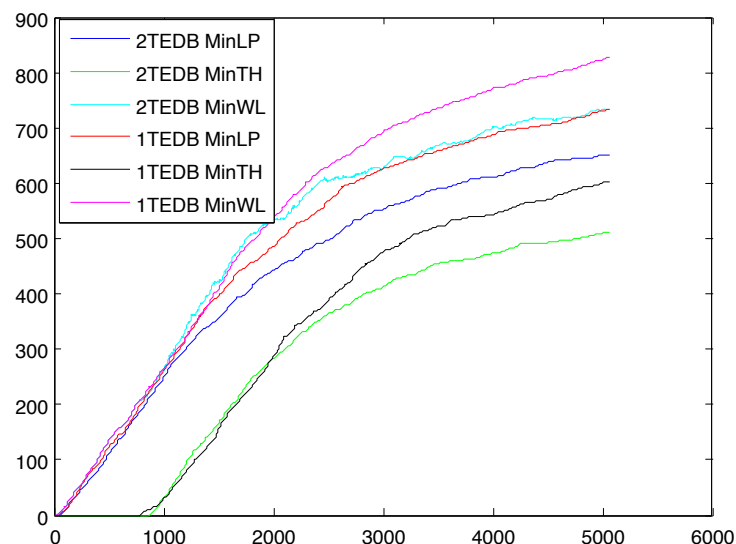


Figura 5-16: Evolución número de operaciones 2 para los distintos casos en simulación estática con 5000 conexiones

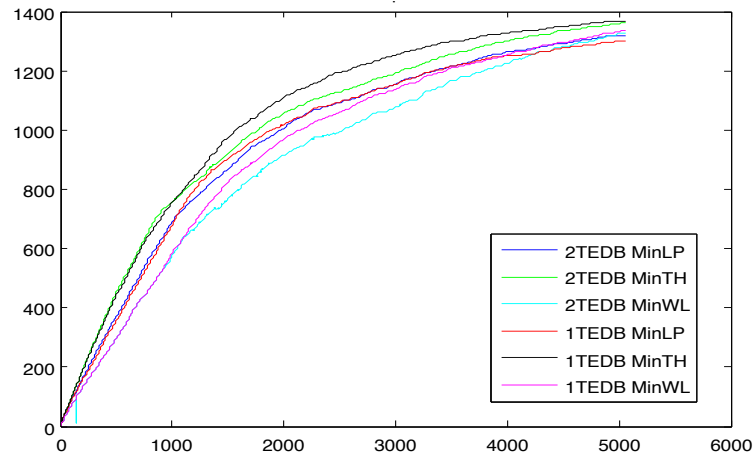


Figura 5-17: Evolución número de operaciones 3 para los distintos casos en simulación estática con 5000 conexiones

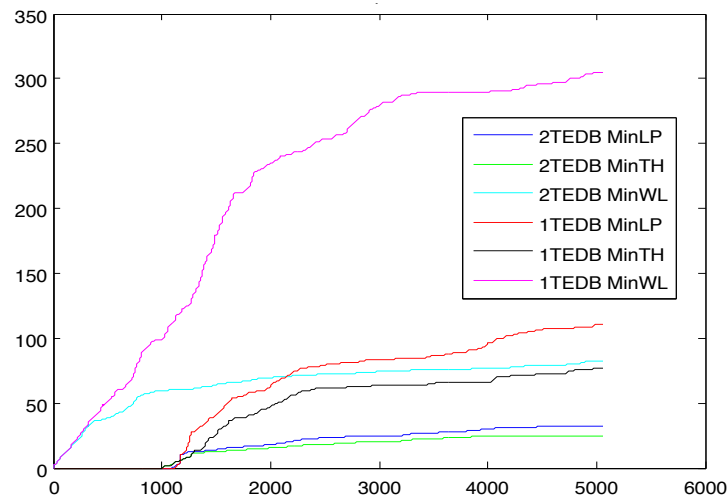


Figura 5-18: Evolución número de operaciones 4 para los distintos casos en simulación estática con 5000 conexiones

Con esto verificamos el funcionamiento de cada política de nuevo. Podemos ver como cada política de rutado hace uso de sus operaciones prioritarias, lo que da lugar a distintas segmentaciones en los recursos de la red.

5.4.2 Comparación de Resultados PCE Multicapa con TEDs separadas y TED multicapa

Los resultados han sido obtenidos con 80 lambdas bidireccionales por cada link de la red y son resultado de simulaciones estáticas, es decir, sólo se establecen *light-paths* que quedan fijos durante toda la simulación. Con ellos obtenemos valores del número de conexiones establecidas y de la cantidad de ancho de banda establecido con las conexiones en toda la red. El tráfico que introducimos en las simulaciones se genera de manera aleatoria entre todos los pares de nodos y a las peticiones les asignamos valores aleatorios de ancho de banda con uniforme entre 1 Gbps y 10 Gbps. Para la comparación de los distintos algoritmos se utilizan patrones idénticos de generación de las muestras, de tal forma que la comprobación sea más real y exacta. Se muestran las dos mismas gráficas de resultados para la red de Telefónica España en la Figura 5-11 y la Figura 5-12.

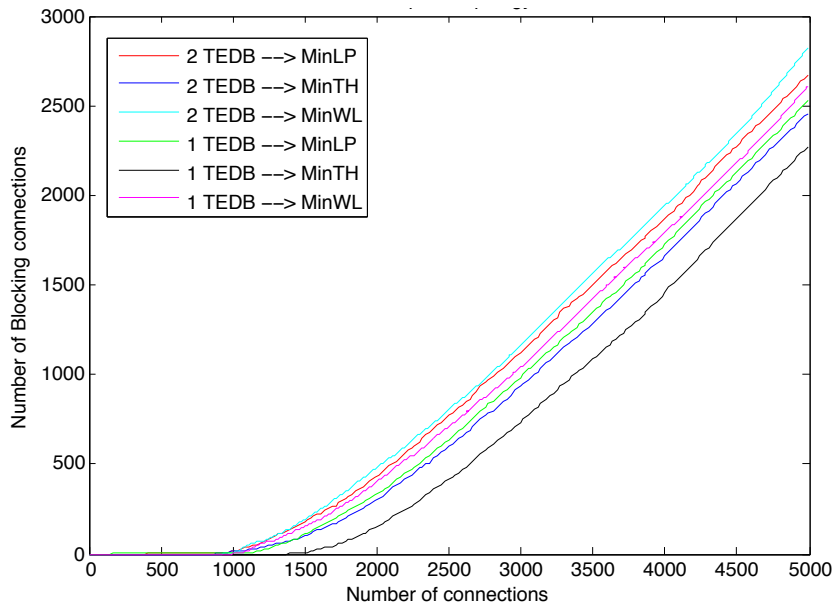


Figura 5-19: Comparación de políticas. Número de conexiones bloqueadas

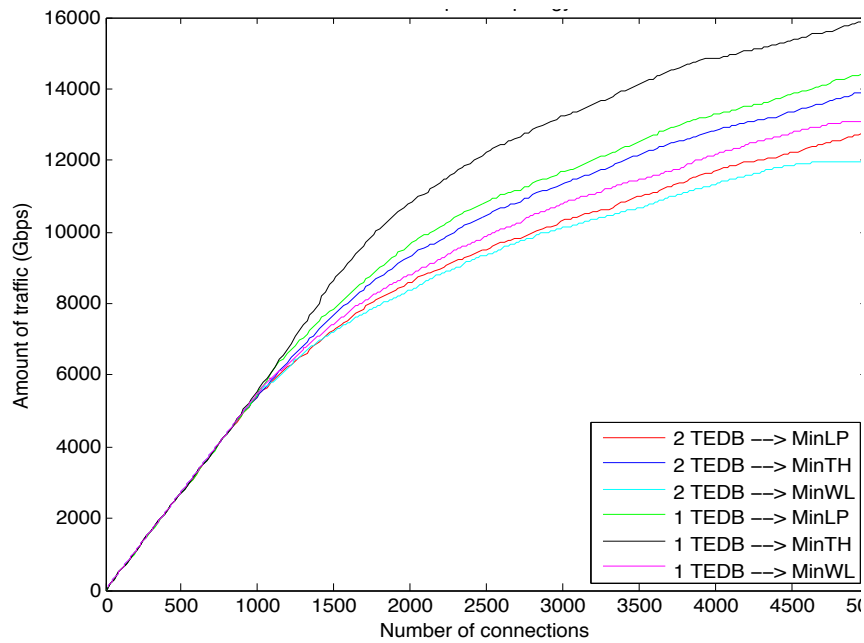


Figura 5-20: Comparación de políticas. Ancho de banda transmitido

En la Tabla 5-3 podemos ver el número de conexiones establecidas en la red multicapa hasta obtener la primera conexión bloqueada, y en la Tabla 5-4 el número total de conexiones bloqueadas en toda la simulación. Vemos como la política que obtiene menor número de bloqueos, y por lo tanto mejor rendimiento en la red, es la MinTH. Para saber de donde salen los valores de la NSFNet 14 tenemos que ir al Anexo A, donde figuran las dos gráficas con los resultados de los algoritmos para esta topología.

	Primera Conexión con bloqueo					
	2 TEDs Separadas			TED Integrada		
	MinTH	MinLP	MinWL	MinTH	MinLP	MinWL
NSFNet-14	831	884	826	1111	923	865

Telefónica Spain	889	962	846	1362	1101	1044
-------------------------	-----	-----	-----	------	------	------

Tabla 5-3: Tabla con las conexiones en las que aparece el primer bloqueo en cada una de las redes y para las 2 políticas de encaminamiento

	Total de conexiones bloqueadas					
	2 TEDs Separadas			TED Integrada		
	MinTH	MinLP	MinWL	MinTH	MinLP	MinWL
NSFNet-14	2917	3006	3095	2856	2993	2945
Telefónica Spain	2453	2661	2804	2269	2516	2550

Tabla 5-4: Tabla con las conexiones totales establecidas en cada una de las redes y para las 2 políticas de encaminamiento

Se puede comprobar como las diferencias entre ambos modelos se incrementan considerablemente cuanto mayor es la red sobre la que realizamos las pruebas. Esto se puede comprobar con los resultados obtenidos en el Anexo A con una red más pequeña (NSF Net 14), ya que se puede apreciar cómo en una red más pequeña las ganancias en términos de optimización son menores. Observando los resultados de la topología de Telefónica España, para el caso de MinTH encontramos la primera conexión bloqueada en la petición 889 en el caso de múltiples TEDs separadas, mientras que para una única TED integrada lo encontramos en la petición 1362. También se hace una comparación en términos cantidad de ancho de banda establecido en la red por las distintas políticas. Podemos ver como para el caso de MinTH se obtiene una mejora del 13.91% en el ancho de banda establecido usando 1 única TED integrada, con respecto al caso de 2 TEDs separadas. En el caso del algoritmo MinLP el incremento en el ancho de banda establecido en la red es de 9.97%.

A continuación se muestran resultados Figura 5-15 para una TED y Figura 5-16 para 2 TEDs, con los algoritmos MinLP y MinTH) sobre la ocupación de longitudes de onda en la capa de transporte al final de las simulaciones para la topología de Telefónica España.

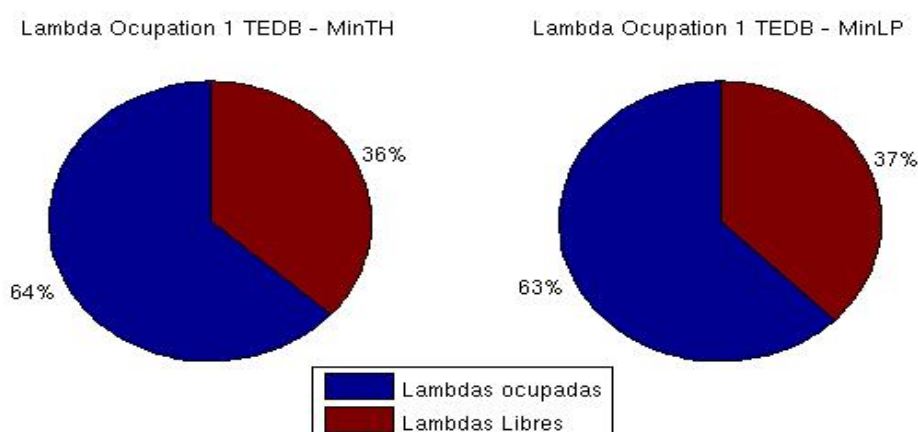


Figura 5-21: Resultados de la ocupación de Lambdas en la red de Telefónica España con 1 TEDs separadas

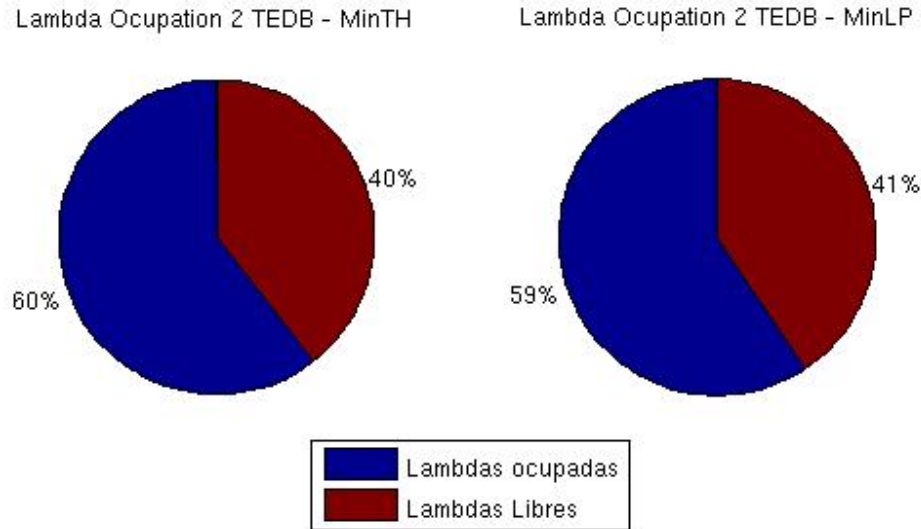


Figura 5-22: Resultados de la ocupación de Lambdas en la red de Telefónica España con 2 TEDs separadas

Por último se añaden unos resultados de rendimiento en los que se han ido variando el número de longitudes de onda (*Lambdas*) que tenemos en cada link bidireccional de la red. De esta manera obtenemos un rendimiento (en esta caso conexiones establecidas / peticiones totales al PCE multicapa). En la Figura 5-17 y la Figura 5-18 se muestran los resultados para los dos modelos de TED. Se puede observar como el algoritmo MinTH es el que mejor rendimiento tiene en la red. También se puede comprobar como en el caso de utilizar una TED Multicapa integrada, los resultados de rendimiento son mucho mejores que el caso de tener la información de red en 2 TEDs separadas.

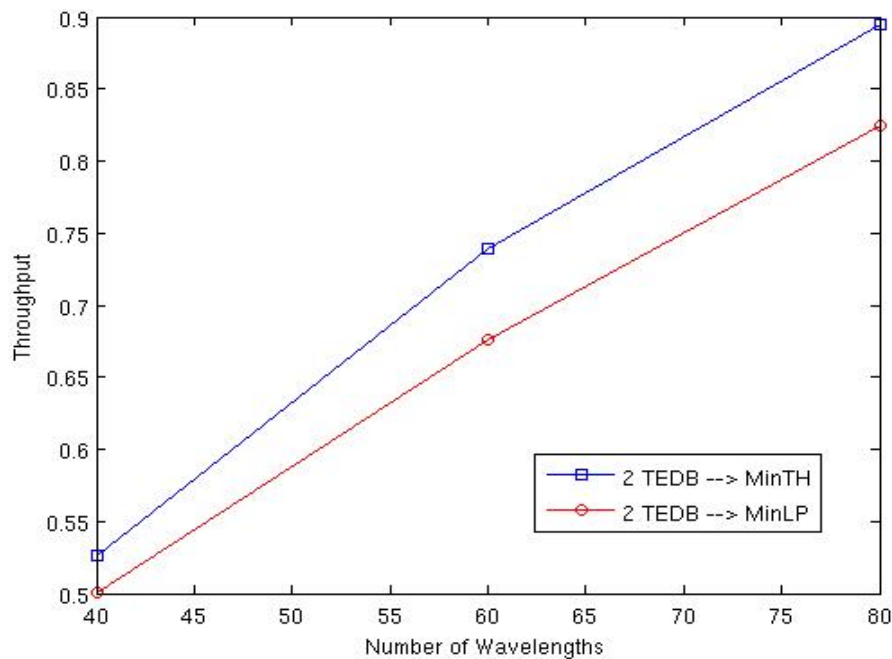


Figura 5-23: Resultados del Rendimiento Variando el número de Lambdas en la red NSFNet14 con 2 TEDs separadas

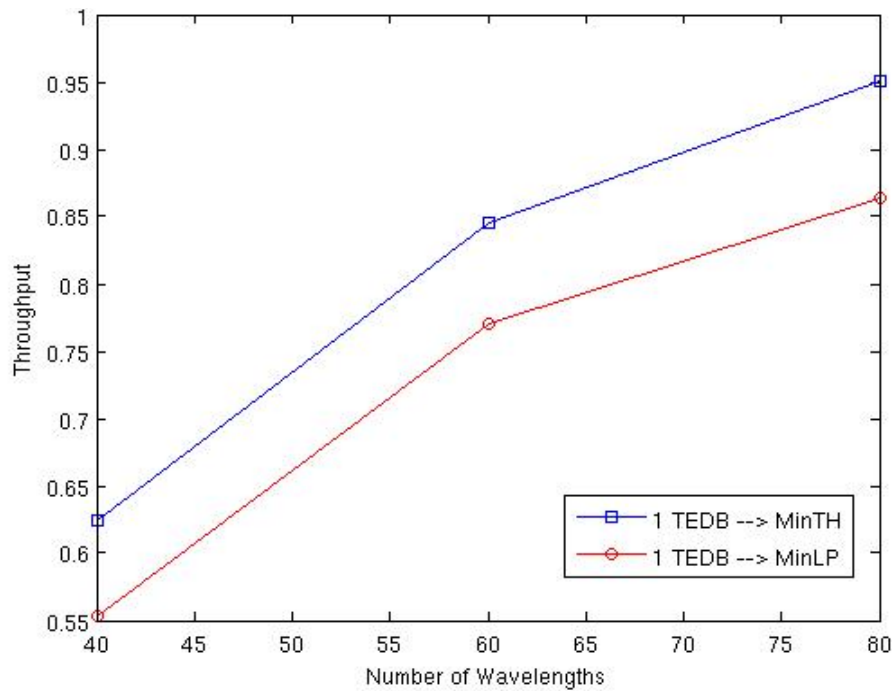


Figura 5-24: Resultados del Rendimiento Variando el número de Lambdas en la red NSFNet14 con 1 TED Multicapa

5.4.3 Tiempos de Computación en PCE Multicapa

En la siguiente gráfica (Figura 5-19) comparamos los resultados obtenidos en los tiempos de computación para los dos modelos de computación y para los algoritmos MinTH y MinLP. La computación de rutas para el caso de PCE Multicapa con TED multicapa completa es de 54.32 ms y 53.57 ms para los algoritmos MinLP y MinTH, respectivamente, mientras que para el caso de PCE con TEDs separadas obtenemos 29.15 ms y 27.19 ms para las mismas políticas. Esto se explica basándonos en los niveles de complejidad de computación para los distintos grafos que tenemos en cada caso. El caso de computación del camino con un grafo multicapa completo hace aumentar el tamaño del mismo, ya que todos los nodos están replicados $W+2$ veces en el grafo. Esto hace que aumenten también los tiempos necesarios para la computación. Para demostrar de manera más clara lo anterior, en el caso de tener una TED integrada, el PCE ha de correr el algoritmo KSP en un grafo considerablemente grande con $N+2$ capas (dónde N es el número de lambdas, 80 en este caso). Como la complejidad del algoritmo KSP aumenta de manera exponencial [45] con el nodos en el grafo, el tiempo de computación para este caso es mucho mayor que en el de 2 TEDs separadas por capa (donde las dimensiones de los 2 grafos son mucho menores).

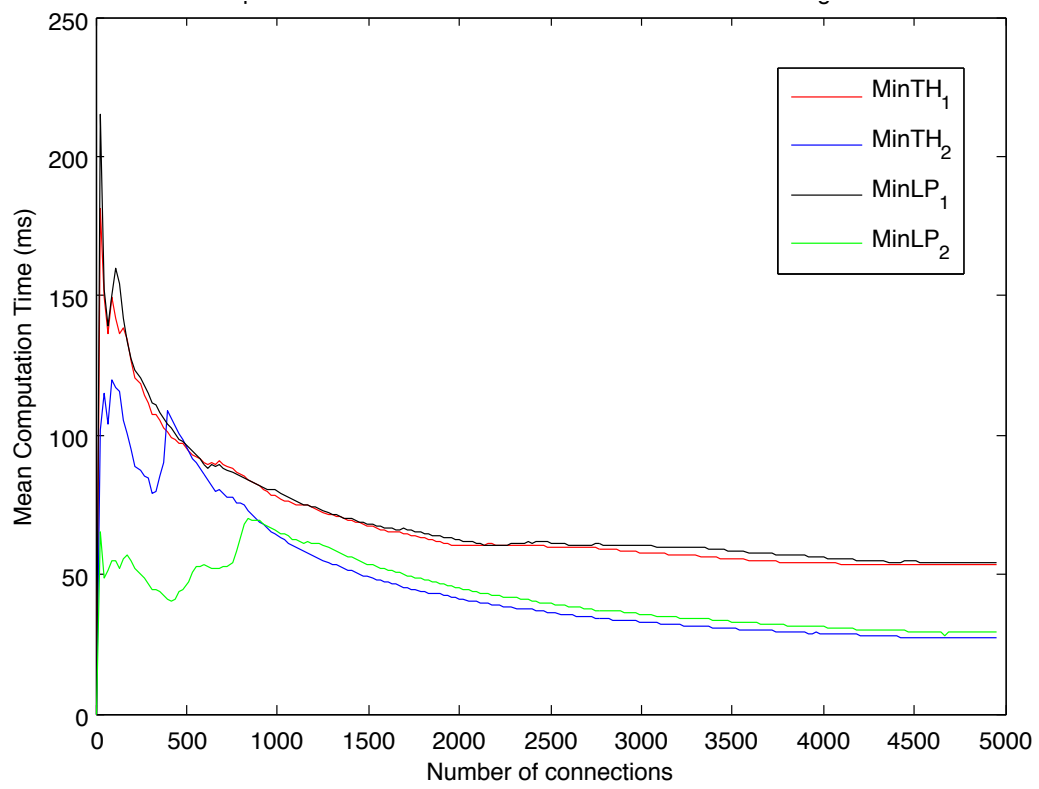


Figura 5-25: Resultados de los tiempos de computación para 2 y 1 TEDs para los algoritmos MinTH y MinLP

6 Conclusiones y futuras líneas de trabajo

En esta sección se van a presentar las conclusiones principales obtenidas del estudio de este proyecto y, posteriormente los aspectos y características que se pueden desarrollar en el futuro para evolucionar sobre los resultados obtenidos.

6.1 Conclusiones

En este proyecto se han desarrollado algoritmos para dar soporte al cálculo de rutas en entornos de PCE para el caso de una red óptica y para una red multicapa con dos modelos: dos TEDs separadas (una por cada capa) una única TED integrada de toda la red. Además se ha extendido el PCE monocapa del testbed de TID para que soporte modelos multicapa.

Se han definido diferentes escenarios para comprobar y evaluar el funcionamiento del PCE multicapa y los distintos algoritmos utilizados. Los modelos de escenarios escogidos son reales y han sido caracterizados con el mayor detalle posible. Estos ejemplos de red utilizados son los siguientes: (1) European Optical Network topology, (2) NSFNet14 y (3) red de Telefónica España.

Sobre estos escenarios se han realizado emulaciones y se muestran los resultados en términos de bloqueo y tiempo de cómputo. De esta forma se trata de demostrar la efectividad de los algoritmos desarrollados para la capa óptica y, posteriormente, multicapa a la hora de rentabilizar el uso de la red. Se pueden obtener dos conclusiones importantes que afectan notablemente al rendimiento en la red en base a los experimentos desarrollados:

- **Almacenamiento de la TED.** La manera de tener almacenada la información sobre nuestra red multicapa, influye en el rendimiento que obtenemos de esta red a la hora de computar los caminos. Podemos observar que cuando tenemos la información de cada capa separada en una TED distinta no le damos al PCE la oportunidad de maximizar los recursos a nivel global, ya que una capa no tiene información complementaria de la otra.
- **Algoritmo en la selección de la longitud de onda.** Otro punto importante es escoger el algoritmo que utilizamos para computar los caminos. Como podemos observar en los resultados tenemos distintas políticas de encaminamiento en base a los criterios de utilización de recursos que tenemos en la red. Según utilicemos un algoritmo u otro podemos obtener distintos rendimientos sobre la red.

Respecto al trabajo realizado desde el punto de vista de implementación, este proyecto final de carrera ha dotado al testbed del CTTC y de TID de algoritmos del estado del arte para operar con redes de conmutación de longitudes de onda. Del mismo modo, ha implementado tres políticas de grooming para entornos multicapa. Este trabajo se ha realizado extendiendo el PCE monocapa de TID para que funcione en entornos multicapa. Gracias a este trabajo en entornos multicapa se ha publicado el artículo *“Assessing the Performance of Multi-Layer Path Computation Algorithms for different PCE Architectures”*, en la conferencia internacional *“Optical Fiber Conference”*. Dicho artículo se muestra en el Anexo C.

6.2 Futuras líneas de trabajo

Se proponen los siguientes puntos como continuación a este trabajo:

- Aplicar políticas y algoritmos de reorganización de los *light-paths* en la capa óptica, también conocidos como desfragmentación. Estos algoritmos ayudarían a optimizar la utilización de los recursos ópticos cada cierto tiempo. La capa de transporte (óptica) normalmente se suele desfragmentar tanto que a menudo quedan muchos recursos (*lambdas*) en desuso, ya que han quedado muy aisladas y no son útiles para el establecimiento de LSPs. Estos algoritmos reorganizan los LSPs establecidos en la capa óptica para que estos recursos no queden aislados.
- Aplicar políticas y algoritmos de borrado o reducción de ancho de banda eficiente de *light-paths* en desuso. Aplicándolo a la liberación de las correspondientes longitudes de onda que dan soporte a esos *light-paths*. Este método es como el anterior, pero afecta a los *light-paths* de la capa IP/MPLS. En simulaciones dinámicas muchos LSPs de la capa superior se quedan establecidos pero con una ocupación mínima sobre su capacidad total. De tal manera que se pueden agrupar algunos *light-paths* para obtener solo uno con mayor capacidad usada sobre el total. Los *light-paths* establecidos tienen una capacidad de 10 Gbps. Si tenemos tres *light-paths* que comparten camino y albergan anchos de banda de 3, 3 y 4 Gbps, podemos agruparlos en un solo *light-path* de manera que tendríamos una capacidad usada de 10 Gbps sobre la capacidad total de 10 Gbps. De manera que podríamos liberar recursos en la capa óptica, reservan solamente los necesarios para dar soporte al único *light-path* que queda establecido.
- Tratar de evitar el robo de ancho de banda obtenido en los resultados aplicando mecanismos de pre-reserva para los LSPs de la capa IP/MPLS. Con ello tendríamos que reservar los recursos de ancho de banda de la capa cliente. El mecanismo sería el mismo que para la reserva de recursos ópticos explicado previamente en la sección 4-3, de tal manera que la creación de este mecanismo no conllevaría mucho problema. Mencionar también que en los resultados obtenidos en este PFC no se ha encontrado mucho bloqueo por este efecto, ya que las simulaciones han sido estáticas y con mucho tiempo entre llegadas, lo que favorece que no se produzcan muchas computaciones simultaneas en el PCE.

Referencias

- [1] L. Berger, IETF, “RFC 3471 - Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description,” January 2003. Online: <http://tools.ietf.org/html/rfc3471>.
- [2] E. Mannie, “RFC 3945 - Generalized Multi-Protocol Label Switching (GMPLS) Architecture”, October 2004. Online: <http://www.ietf.org/rfc/rfc3945>.
- [3] ITU, “Rec. G.8080/Y.1304: Architecture for the automatically switched optical network (ASON),” February 2005.
- [4] D. Papadimitriou, “RFC 4328 - Generalized Multi-Protocol Label Switching (GMPLS) Signaling Extensions for G.709 Optical Transport Networks Control”, January 2006. Online: <http://www.ietf.org/rfc/rfc4328>
- [5] K. Shiomoto, D. Papadimitriou, J.L. Le Roux, M. Vigoureux, D. Brungard, “Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)”, July 2008. Online: <http://tools.ietf.org/html/rfc5212>
- [6] Ning Zhang, Hong Bao, “GMPLS Technology and Its Application in WDM Optical Network”, 2009. In International Conference on Networking and Digital Society.
- [7] R. Muñoz, R. Martínez, R. Casellas, R. Morro, C. Cavazzoni, S. Pizzaja, M. Jaeger, H.-M. Foisel, J. Jiménez, C. García, H. Dentler, “Experimental Demonstrations of ASON-GMPLS signaling interworking in the NOBEL2 Multi-domain Multi-layer Control Plane Emulator”, 2008. In ONDM.
- [8] A. Farrel, J.-P. Vasseur, J. Ash, “RFC – 4655 A Path Computation Element (PCE)-Based Architecture”, August 2006. Online: <http://www.ietf.org/rfc/rfc4655>
- [9] JP. Vasseur, J.L. Le Roux, “Path Computation Element (PCE) Communication Protocol (PCEP)”, March 2009. Online: <http://tools.ietf.org/html/rfc5440>
- [10] J. Ash, J.L. Le Roux, “Path Computation Element (PCE) Communication Protocol Generic Requirements”, September 2006. Online: <http://tools.ietf.org/html/rfc4657>
- [11] J.L. Le Roux, “Requirements for Path Computation Element (PCE) Discovery”, October 2006. Online: <http://www.ietf.org/rfc/rfc4674>
- [12] E. Oki, T. Takeda, J.L. Le Roux, A. Farrel, “Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering”, September 2009. Online: <http://tools.ietf.org/html/rfc5623>
- [13] V. López, B. Huiszoon, J. Fernández-Palacios, O. González de Dios, J. Aracil, “Path Computation Element in Telecom Networks: Recent Developments and Standardization Activities”. In ONDM 2010.
- [14] T. Takeda, “Framework and Requirements for Layer 1 Virtual Private Networks”, April 2007. Online: <http://tools.ietf.org/html/rfc4847>
- [15] José Enríquez Gabeiras, Víctor López, Javier Aracil, Juan Pedro Fernández-Palacios, Carlos García Argos, Óscar González de Dios, Francisco Javier Jiménez Chico, José Alberto Hernández. “Is multilayer networking feasible?”, in Elsevier Journal on Optical Switching and Networking, April 2009.

- [16] Martin Nathansen, Ulrich Häbel, Cyril Margaria, Elie Sfeir, Andreas Iselt, Michael Bauer, “Implementation of UNI in optical multi-layer networks”, 2006 ITG Symposium on Photonic Networks.
- [17] Víctor López, Óscar González de Dios, José Alberto Hernández, Raúl Duque, Carlos García Argos, Javier Jiménez Chico, Juan Pedro Fernández-Palacios, Javier Aracil, “Performance evaluation of Threshold-based Multi-layer Traffic Engineering strategies”, 2009. In NOC 2009.
- [18] Javier Jimenez, Óscar González, Carlos García, Juan Pedro Fernández-Palacios, María Ángeles Callejo, Javier Antich, Luc Ceuppens, Michael Roth, “An Industrial Experience on IP/MPLS and Transport Layer Coordination”, IEEE Communications Magazine, Accepted for Review Nov. 2010.
- [19] K. Matsui, T. Sakurai, M. Kaneda, J. Murayama, and H. Ishi, “A multilayered traffic engineering architecture for the electronic/optical hybrid network,” in Processing IEEE Pacific Rim Conference on Communications, Computers and Signal, PACRIM, vol. 1, Aug 2003, pp. 293–296.
- [20] J. Comellas, R. Martinez, J. Prat, V. Sales, and G. Junyent, “Integrated IP/WDM routing in GMPLS-based optical networks,” IEEE Network, vol. 17, no. 2, pp. 22–27, mar 2003.
- [21] D. Zhemin, M. Hamdi, J. Y. B. Lee, and V. O. K. Li, “Integrated routing and grooming in GMPLS-based optical networks,” in IEEE International Conference on Communications, vol. 3, Jun 2004, pp. 1584–1588.
- [22] S. De Maesschalck, M. Pickavet, D. Colle, and P. Demeester, “Multi-layer traffic grooming in networks with an IP/MPLS layer on top of a meshed optical layer,” in Global Telecommunications Conference, IEEE GLOBECOM, vol. 5, Dec 2003, pp. 2750–2754.
- [23] Martin Vigoureux, Bela Berde, Loa Andersson, Acreo, Tibor Cinkler, Lieven Levrau, Marconi Ondata, Didier Colle, Juan Fdez-Palacios, Monika Jäger, “Multilayer Traffic Engineering for GMPLS-Enabled Networks”, July 2005, in IEEE Communications Magazine.
- [24] Jie Zhang, “Research on Unified Control Plane Design for Multi-layer Optical networks”, 2010. At IEEE
- [25] Emmanuel Dotaro, Martin Vigoureux, Dimitri Papadimitriou, “Multi-Region Networks: Generalized Multi-Protocol Label Switching (GMPLS) as enabler for vertical integration”, 2004. Globecom 2004 Workshops.
- [26] J. Lang, IETF, “RFC 4204 - Link Management Protocol (LMP)”, October 2005. Online: <http://tools.ietf.org/html/rfc4204>.
- [27] A. Fredette, J. Lang, “RFC 4209 - Link Management Protocol (LMP) for Dense Wavelength Division Multiplexing (DWDM) Optical Line Systems”, October 2005. Online: <http://tools.ietf.org/html/rfc4209>.
- [28] V. Beeram, J. Drake, “GMPLS-UNI BCP draft-beeram-ccamp-gmpls-uni-bcp-01.txt”, March 12, 2012. Online: <http://tools.ietf.org/html/draft-beeram-ccamp-gmpls-uni-bcp-01>.
- [29] G. Swallow, J. Drake, H. Ishimatsu, Y. Rekhter, “RFC 4208 Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource

- ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model”, October 2005. Online: <http://www.ietf.org/rfc/rfc4208>.
- [30] Huawei, “Technical White Paper for the Unified Control Plane of the SingleBackbone Solution”, 2010.
 - [31] Ahmed Mokhtar, Member, IEEE, and Murat Azizoglu, Member, IEEE, “Adaptive Wavelength Routing in All-Optical Networks”, 1998. IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 6, NO. 2, APRIL 1998.
 - [32] Raúl Muñoz, Ramón Casellas, Ricardo Martínez, “An Open GMPLS-enabled Control Plane testbed for remote development and experimentation of PCE-based path computation algorithms”, 2010. At Future Network Summit 2010.
 - [33] Hongyue Zhu, Hui Zang, Keyao Zhu, Biswanath Mukherjee, “A Novel Generic Graph Model for Traffic Grooming in Heterogeneous WDM Mesh Networks”, 2003. IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 11, NO. 2, APRIL 2003.
 - [34] K. Kompella, Y. Rekhter, “Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)”, October 2005. Online: <http://tools.ietf.org/html/rfc4206>
 - [35] Adrian Farrel and Igor Bryskin, “GMPLS: Architecture and Applications”.
 - [36] K. Kompella, Y. Rekhter, “Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)”, October 2005. Online: <http://www.ietf.org/rfc/rfc4202>
 - [37] K. Kompella, Y. Rekhter, “OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)”, October 2005. Online: <http://tools.ietf.org/html/rfc4203>
 - [38] R. Braden, “Resource ReSerVation Protocol (RSVP)”, September 1997. Online: <http://www.ietf.org/rfc/rfc2205>
 - [39] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, “RSVP-TE: Extensions to RSVP for LSP Tunnels”, December 2001. Online: <http://tools.ietf.org/html/rfc3209>
 - [40] Daisaku Shimazaki, Eiji Oki, Kohei Shiimoto and Naoaki Yamanaka, “GMPLS and IP+MPLS Interworking Technologies Routing and Signaling”, 2004.
 - [41] Bart Puype, Qiang Yan, Didier Colle, Sophie De Maesschalck, Ilse Lievens, Mario Pickavet, Piet Demeester, “Multi-layer Traffic Engineering in Data-centric Optical Networks”, 2003. At ONDM 2003.
 - [42] V. López, J.P. Fernández-Palacios, M. Izal, “Implementation of Multi-layer Techniques Using FEDERICA, PASITO and One Lab Network Infrastructures”, at ICON 2011.
 - [43] L. Berger, “Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions”, January 2003. Online: <http://tools.ietf.org/html/rfc3473>
 - [44] O. González de Dios, “PCEP Extensions for Temporary Reservation of Computed Paths Resources and Support for Limited Context State in PCE”, October 2010. Online: <http://tools.ietf.org/html/draft-gonzalezdedios-pce-reservation-state-00>

- [45] Feijo, "Algoritmo de Dijkstra", Online:
http://prof.usb.ve/bfeijoo/dat/CO5422/200909_CO5422_Clase_16.pdf

Glosario

API	Application Programming Interface
ASON	Automatically Switched Optical Network
ATM	Asynchronous Transfer Mode
BR	Border Router
CSPF	Constrained Shortest Path
CTTC	Centre Tecnològic de Telecomunicacions de Catalunya
DmxE	Demux Edge
DWDM	Dense Wavelength Division Multiplexing
E2E	End-to-End
E-NNI	External Network to Network Interface
ERO	Explicit Route Object
FA	Forwarding Adjacency
FF	First Fit
FSM	Finite State Machine
GMPLS	Generalized Multiprotocol Label Switching
GrmE	Grooming Edge
H-LSPs	Hierarchical LSPs
I-NNI	Internal Network-to-Network Interface
IP	Internet Protocol
KSP	K-Shortest-Path
L2SC	Layer Two Switch Capable
LER	Label Edge Router
LMP	Link Management Protocol
LPE	Light-path Edge
LSA	Link State Advertisement
LSC	Lambda Switch Capable
LSP	Label Switched Path
LSR	Label Switch Router
ML	Multi Layer
MinLP	Minimizing the Number of Light-paths
MinTH	Minimizing the Number of Traffic Hops

MinWL	Minimizing the number of Wavelength-Links
MPLS	Multiprotocol Label Switching
MuxE	Mux Edge
NMS	Network Management System
NNI	Network to Network Interface
OF	Objective Function
OPEX	Operational Expenditure
OSNR	Optical to Signal Noise Ratio
OSPF	Open Shortest Path First
OTN	Optical Transport Network
OXC	Optical Cross Connect
PB	Probabilidad de bloqueo
PCC	Path Computation Client
PCE	Path Computation Element
PCEP	Path Computation Element Communication Protocol
PFC	Proyecto Fin de Carrera
PR	Pre-Reserva
PSC	Packet Switch Capable
QoS	Quality of Service
RFC	Request for Comments
RP	Request Parameters
RSVP	Resource ReSerVation Protocol
RWA	Routing and Wavelength Assignment
RxE	Receiver Edge
SLA	Service Level Agreements
SONET	Synchronous Optical Network
TCP	Transmission Control Protocol
TDM	Time-Division Multiplexing
TE	Traffic Engineering
TED	Traffic Engineering Database
TID	Telefónica I+D
TxE	Transmitter Edge
UNI	User Network Interface

UCP	Unified Control Plane
WBE	Wavelength Bypass Edge
WDM	Wavelength-Division Multiplexing
WL	Wavelength Link
WLE	Wavelength-Link Edge
WSON	Wavelength Switched Optical Networks

Anexos

A Resultados Multicapa NSFNet 14

En esta sección introducimos los resultados multicapa obtenidos con la red NSF Net 14. En la Figura A-1 se muestran los resultados de peticiones bloqueadas en la red. Y en la Figura A-2 se muestran los resultados de ancho de banda establecido por cada algoritmo.

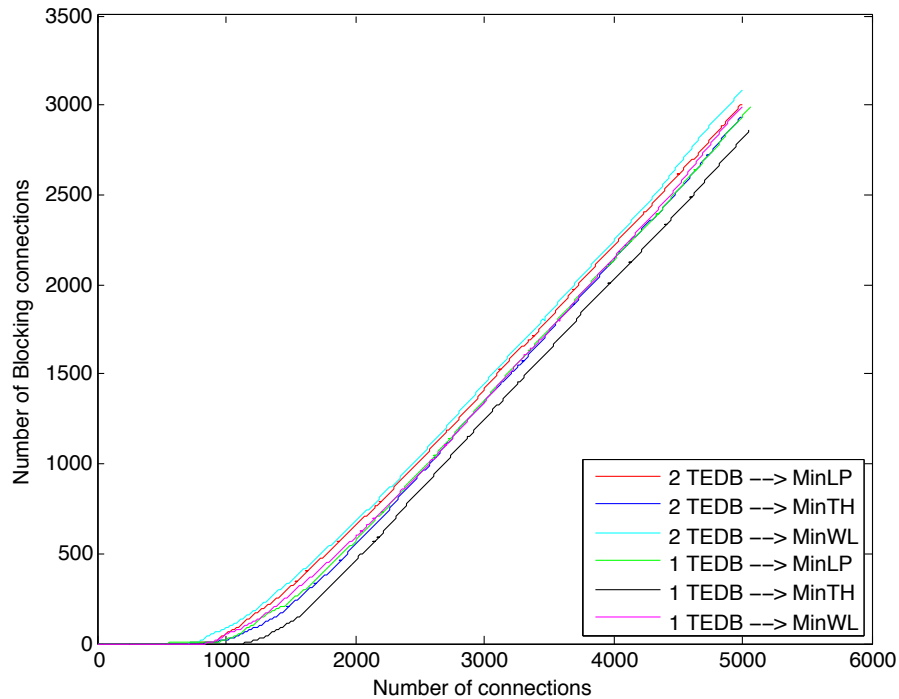


Figura A-1: Ejemplo Registro del algoritmo en el PCEServerConfiguration.xml

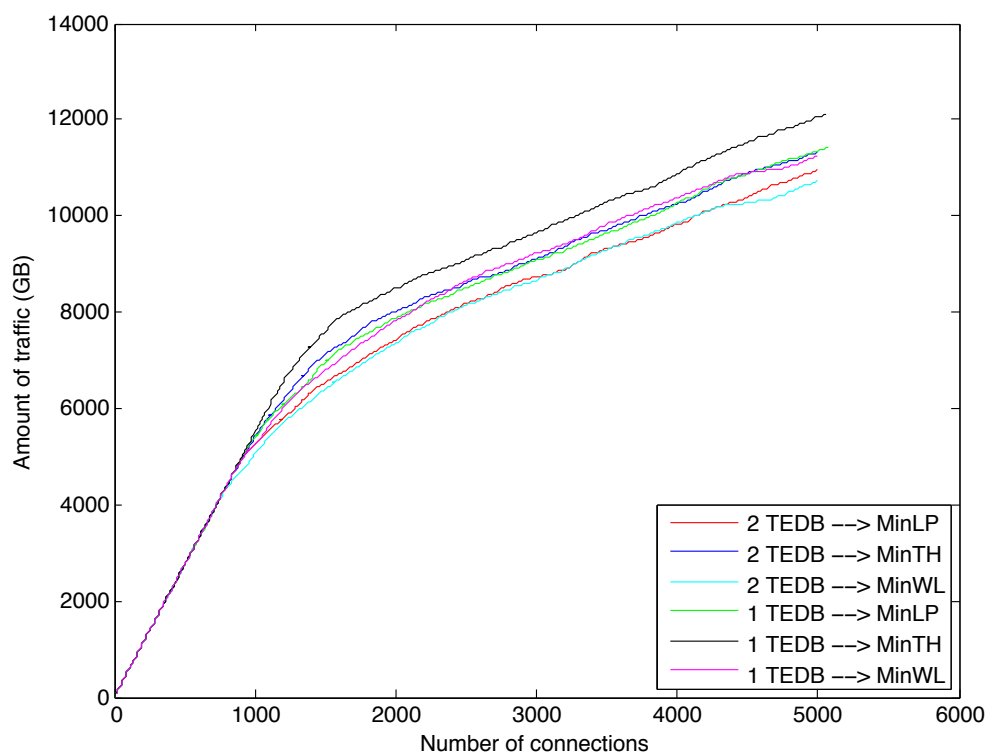


Figura A-2: Ejemplo Registro del algoritmo en el PCEServerConfiguration.xml

Podemos comparar estos resultados con los de la topología de España para comparar la validez y los resultados de los algoritmos.

B Manual del Usuario

Información previa

Para la realización de pruebas necesitamos disponer al menos de 2 servidores con sistema operativo LINUX. Un servidor los utilizaremos para simular el PCE Server. En este servidor tenemos que situar los siguientes archivos:

- Archivo XML con la topología de red para la simulación.
- PCEServerConfiguration.xml: Archivo con los parámetros de configuración del PCE. Los principales parámetros que introducimos son los siguientes:
 - Un puerto para la simulación.
 - Un puerto de gestión.
 - Definir un puerto para la sesión OSPFTCP.
 - Nombre del archivo XML con la topología.
 - *Flag* multicapa: true/false.
 - *Flag* pre-reserva: true/false.
 - *Flag* para indicar TED única o 2 TEDs → isCompletedAuxGraph: true/false.
 - *ID Grooming policy* → idAlgo:
 - 0: Minimizing the number of Traffic Hops.
 - 1: Minimizing the number of *light-paths*.
 - 2: Minimizing the number of Wavelength Links.
 - Nombre del fichero java con el nombre del algoritmo para registrarlo asignándole un código OF. Ej.:

```
<algorithmRule of="1100" svec="false" name="multilayer.Multilayer_MinTH_Algorithm" isParentPCEAlgorithm="false" isWSNAlgorithm="true"/>
```

Figura B-3: Ejemplo Registro del algoritmo en el PCEServerConfiguration.xml

- PCE.jar: ejecutable con el código JAVA para realizar las pruebas.
- pce.sh: script para lanzar el PCE Server.

En otro servidor cargaremos el Cliente (NMS) y el VNTM, con puertos diferentes. En el Cliente tenemos que incluir los siguientes archivos:

- Archivo XML con la topología de red para la simulación.
- AutomaticTester.xml: archivo que contiene los parámetros de configuración del Cliente y de la simulación. Los principales parámetros a introducir son los siguientes:
 - Dirección IP y puerto donde estamos simulando el PCE Server.
 - Definir un puerto de gestión para el Cliente.
 - Definir los tiempos para la simulación, ambos en milisegundos. Estos tiempos son los que nos definen la carga que le metemos a la red en la simulación:
 - MeanTimeBetweenRequest (λ): tiempo global medio entre peticiones. Para cada *meanTimeBetweenRequest* introducimos $N \times (N-1)$ peticiones en la simulación, donde N es el número de nodos que tenemos en la red.

- MeanConectionTime (μ): tiempo de conexión medio.
- *Flag* para indicar (en caso de simulación multicapa o no) si tenemos sesión VNTM: true/false.
- Definir el número de nodos que tiene nuestra topología para poder calcular la carga de la red de la siguiente manera:

$$\rho = \frac{\mu \times ((N - 1) \times N)}{\lambda}$$
- Definir el código OF del algoritmo que queremos utilizar.
- *Flag* para indicar si realizamos pre-reserva: true/false.
- En caso de tener pre-reserva hay que definir el tiempo de reserva.
- Definir el rango de ancho de banda que van a tener las peticiones. Sigue una distribución aleatoria.
- NetworkEmulatorConfiguration.xml: archivo para la configuración del emulador de red. Definimos lo siguiente:
 - Parámetros del PCE Server (IP y puerto) para definir la sesión OSPF.
 - *Flag* para indicar si la simulación es multicapa: true/false.
 - Nombre del archivo .xml con la topología de red que vamos a simular.
 - Indicamos el tipo de emulador de red que utilizamos.
- PCE.jar: ejecutable con el código JAVA para realizar las pruebas.
- Client.sh: script para lanzar el Cliente (NMS) en el servidor.

Por último tenemos el VNTM, que lo ejecutamos en el mismo servidor que el NMS pero con distintos puertos. Tenemos que incluir los siguientes archivos:

- Archivo XML con la topología de red para la simulación.
- VNTMConfiguration.xml: contiene la dirección del servidor donde ejecutamos el VNTM, así como los puertos que le damos (siempre distintos a los del NMS).
- NetworkEmulatorConfiguration.xml: similar al que tenemos para el Cliente.
- PCE.jar: ejecutable con el código JAVA para realizar las pruebas.
- VNTMServer.sh: script para lanzar el VNTM en el servidor.

Ejecución

Con todos estos ficheros podemos lanzar las ejecuciones. Primero tenemos que lanzar el PCE Server, después el VNTM y por último el Cliente. Si lo realizamos en otro orden nos mostrará un error por pantalla. Para ir ejecutando las distintas partes no tenemos más que ir introduciendo las siguientes instrucciones:

- Desde el PCEServer:

```
ale@foton1:~/Sergio/PCEServer_2_TEDBs_MinLP$ sudo /usr/local/pce/bin/dpce.sh
```

- Desde el Cliente:

```
maggie@maggie:~/Sergio/Client/Client_MULTILAYER_2_TEDs_MinLP$ sudo ./client.sh
```

- Desde el VNTM:

```
maggie@maggie:~/Sergio/VNTM_2_TEDs_MinLP$ sudo ./VNTMServer.sh
```

Una vez empezada la ejecución podemos ir analizando el estado de la topología (TED) a través del puerto de mantenimiento del PCE de la siguiente manera:

```
ale@foton1:~/Sergio/PCEServer_2_TEDBs_MinLP$ telnet localhost 7989
Trying ::1...
Connected to localhost.
Escape character is '^]'.
PCE:>help
Available commands:
show topology
quit
PCE:>
```

Figura B-4: Utilización del puerto de gestión

Si introducimos el comando “*show topology*” podemos ver el estado de los links de toda la red. En el caso de la capa de transporte podemos ver los “*bit maps*” de utilización de las longitudes de onda de los links ópticos.

Resultados

Los resultados se van guardando en un archivo *.txt* en la misma carpeta donde tenemos los archivos del Cliente.

C Publicaciones Relacionadas

Como resultado de este PFC se ha escrito el siguiente artículo:

- *Sergio Martínez, Víctor López, Mohit Chamania, Oscar González de Dios, Admela Jukan y Juan Pedro Fernández-Palacios, “Assessing the Performance of Multi-Layer Path Computation Algorithms for different PCE Architectures”, en Proc. Optical Fiber Conference (OFC) 2013, Mar. 2013.*

Ha sido enviado, aceptado y posteriormente publicado en el siguiente congreso internacional:

*Optical Fiber Communication Conference 2013
36th International Conference
March 17-21, 2013
Anaheim, California (USA)*

Se puede obtener más información del congreso en su página web:

<http://www.ofcnfoec.org/home/>

El artículo está indexado por OSA (Optical Society (of America)) y el IEEE. Y se puede encontrar en la siguiente URL:

<http://www.opticsinfobase.org/abstract.cfm?URI=OFC-2013-OW4G.1>

A continuación se muestra la versión final del artículo.

Assessing the Performance of Multi-Layer Path Computation Algorithms for different PCE Architectures

S. Martínez^{1*}, V. López¹, M. Chamania², O. González¹, A. Jukan², J.P. Fernández-Palacios¹

¹Telefónica I+D, Madrid, Spain ²Technical Technische Universität Carolo-Wilhelmina zu Braunschweig

*e-mail: smta@tid.es

Abstract: We have implemented a multi-layer PCE and compared performance of various algorithms using either one integrated, or two separate PCEs in each layer. Multi-layer integration reduces blocking by 13,91%, but increases computation time by 49,24%.

OCIS codes: (060.0060) Fiber optics and optical communications; (060.4256) Networks, network optimization.

1. Introduction

To address the increasing service demand in an efficient fashion, operators are considering the multi-layer architectures, with IP routing and optical switching working together [1]. A typical core network consists of an IP/MPLS network designed over a Wavelength-Switched Optical Network (WSO) network. In a combined network infrastructure, the IP/MPLS links are made up of Lambda switching Capable (LSC) Label Switched Paths (LSPs) established in the transport network based on optical transmission, while connections/services are established using Packet Switching Capable (PSC) LSPs in the IP/MPLS network. Here, a joint path computation is essential, which can be implemented within the already existing Path Computation Element (PCE) framework.

There are basically two different schemes proposed to compute multi-layer paths with the PCE [2]: (i) integrated (single) multi-layer PCE where a unique PCE has the complete multi-layer topology information, a centralized PCE server uses topology information available in both Traffic Engineering Databases (TEDBs) to compute optimal paths. (ii) coordinated (multiple) PCEs solution where each network layer has its own PCE that can communicate with each other when required to compute multi-layer paths. The exchange of requests allows the IP/MPLS network layer to request a path segment in the optical (server) layer to optimize the provisioning of one or more connections requested in the client layer. In our previous work [3], we presented an implementation of the multiple cooperating PCEs, and in this work we implement the single multi-layer PCE and various path computation algorithms, such as proposed in [4], for both integrated and coordinated PCE solutions to compare the performance in terms of blocking and computation delay for both PCE architectures.

2. The Multi-Layer Algorithms Implemented to Test

The primary challenge of multi-layer path computation algorithms is to set up new LSPs in different layers while taking into account different layer-specific constraints. Multi-layer algorithms in a typical network topology should be able to (1) route lightpaths over the physical topology (WSO layer), by calculating new LSPs, and assign wavelengths to generate the client network topology in the IP layer, and (2) route connections in the client network topology. According to [5], multi-layer path computation for a request in the client network can be reduced to a combination of one or more of 4 different operations: (1) Operation 1 attempts to route a connection in the client network over an existing lightpath directly between the source and the destination. This operation is equivalent to computing a 1-hop path in the client network for a request. (2) Operation 2 attempts to allocate the request in the client network only, i.e. by only using existing lightpaths. (3) Operation 3 creates a new lightpath from source to destination to route the request, and in this operation an LSP is computed in the transport network topology from the

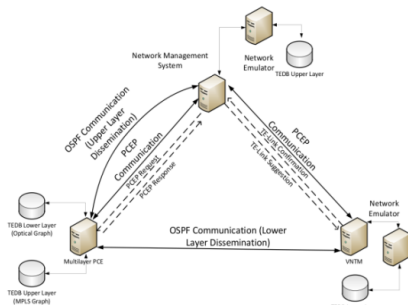


Fig. 1. Telefónica emulator architecture with ML-PCE

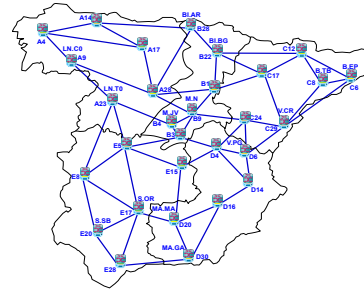


Fig. 2. Telefónica Spanish Backbone Network

source to the destination. (4) Finally, operation 4 routes the traffic over existing lightpaths and creates one or more lightpaths between intermediate nodes to route a connection from a source to a destination. Based on these operations, authors in [5] identify three different policies for multi-layer path computation:

- **Minimizing the number of Traffic Hops (MinTH).** This policy firstly uses Operation 1 to compute an existing single-hop path and if it fails, a new end-to-end lightpath is created (Operation 3). If both fail, paths are computed using Operations 2 and 4 and the one with fewer hops on the virtual topology is chosen.
- **Minimizing the number of Light-Paths (MinLP).** The policy first uses Operation 1, and if unsuccessful Operation 2 is run to compute a path in the virtual topology. In case of failure, Operation 3 and later Operation 4 are performed to compute a path.

3. Testbed set-up

In our testbed set-up, we compare the performance of the integrated multi-layer PCE and the coordinated multi-layer PCE framework with a real world PCE implementation (via emulations). Fig. 1 shows the reference architecture for the integrated multi-layer PCE scenario. The physical network used is the Spanish Backbone Network (Fig. 2), with each link supporting 80 wavelengths (realistic assumption). We implement basic network emulators to maintain reservation status, statistics and update the TEDs. A Network Management System (NMS) generates 5000 connection requests with random source-destination selection and random bandwidth selection between 1 and 10Gbps. For the purpose of this study, all connections, once established, are assumed to be permanent to check the PCE performance and grooming behavior in a non-dynamic scenario. The request from the NMS is sent to the ML-PCE which can compute single or multi-layer paths. The multi-layer path is sent in the form of a multi-layer ERO, an example of which is Fig. 6. The multi-layer ERO is constructed using standard ERO objects to represent paths segments in individual layers, and use the SERVER_LAYER_INFO sub-object to indicate a layer change. For the example shown in Fig. 6, the path is computed as an end-to-end lightpath (Operation 3) from a source (192.168.8.3) to destination (192.168.8.1) in the MPLS network. After the source node, a SERVER_LAYER_INFO sub-object is included to highlight the layer change. The next sub-objects are unnumbered interfaces as well as Label control, which identify the lambda in the optical domain. Once the optical path is defined in the ERO, another SERVER_LAYER_INFO sub-object is included before the destination to identify the change to the upper layer. In case of a multi-layer path, the NMS must first establish the lightpaths before establishing the connections, and requests the Virtual Network Topology Manager (VNTM) for the same by forwarding the ERO inside a TE-Link Suggestion message to the VNTM [5]. If the lightpaths are established successfully, the NMS and the VNTM send OSPF LSA messages to the PCE to update the PCE TED. For the multiple PCE scenario, all other elements remain unchanged but instead of a single PCE, there are 2 PCEs. The NMS sends a request to the MPLS PCE, which if required communicates with the WSON PCE to compute a path. The computed WSON LSPs are then included by the MPLS PCE in a multi-layer ERO and sent to the NMS.

To run the aforementioned policies in a single TED, authors in [4] propose to construct an auxiliary graph with $N+1$ layers, where N is the number of wavelengths (80 for this experiments). The edge weights in this auxiliary graph are defined for each policy [4] and the K-Shortest Path (KSP) algorithm is used to compute a path. For the two TEDBs case, Operations 1,2 and 3 can be computed by individual PCEs (MPLS or WSON) while for Operation 4, the MPLS PCE generates suggestions for ingress and egress MPLS routers between which an LSP should be established, and the WSON PCE then attempts to compute a lightpath between the suggested ingress-egress nodes.

4. Algorithms evaluation

Fig. 3 shows the number of blocked connections for MinLP and MinTH policies in the case of a single or two TEDBs. For any policy, we see that a single TEDB reduces the number of blocked connections in the network. For example, in the case of the MinTH policy, the first blocked connection is observed at request 889 when multiple PCEs are used and at request 1362 for a single PCE case, while the corresponding numbers for the MinLP policy are 962 and 1101 connections respectively. Fig. 4 shows the allocated bandwidth for both policies in the case of a single or multiple PCEs, and we see that the MinTH policy can improve total capacity allocation by 13.91% when using a single PCE as compared to multiple PCEs while the increase in the MinLP policy is 9.97%.

The integrated PCE contains the complete topology information and as a result can optimize resource allocation for any algorithm, when compared with the multiple PCE. The coordinated PCE on the other hand increases the utilization of the optical resources in the network: the MinTH policy uses 64,3% and 60,2% of the wavelengths for single and multiple PCE scenario respectively, while the same values for the MinLP are 63,1% and 58,9%, due to the preference to minimize new lightpaths.

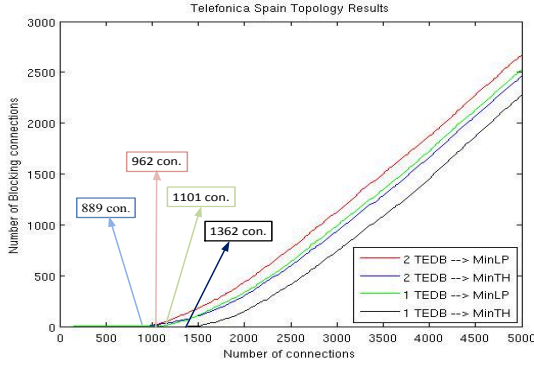


Fig. 3. Number of blocking connections

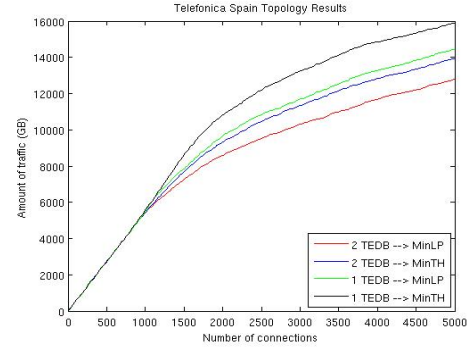


Fig. 4. Amount of allocated bandwidth

We also compare the mean computation time taken for both policies in case of single and Multiple PCEs in Fig. 5. The computation on the Single PCE has to work with a large topology and as a result needs 54.32ms and 53.57ms for MinLP and MinTH policies, respectively, while the multiple PCE solution computes connections for the same policies in 29.15ms and 27.19ms respectively, as the computation of the paths in individual layers have significantly lowered complexity than the integrated multi-layer path computation. Note that both PCEs are implemented on the same host for this specific scenario.

In light of the results, we can conclude that there is an important increment in computation time for integrated multi-layer PCE solution. Here, the PCE is running KSP in a large graph, which has $N+1$ layers (81 in this case). As KSP complexity increases linearly with number of arcs and nodes the computation time for this algorithm is much higher than in the case of multiple PCEs, where each of PCEs runs their own KSP is run. Note that in the case of multiple coordinated PCEs, First Fit is run for wavelength assignment once KSP finds candidate paths.

5. Conclusions

Multi-layer coordination is becoming an important requirement in network operators, due to its potential to reduce carrier's investments, network interventions and thus cost of operations. Multi-layer PCE is a key technology to study for this scenario. This work shows the performance achieved with full topological information or with partial information. To do so, we have implemented a state-of-the-art protocol in Telefonica PCE test-bed and compare the performance. Results show that integrated TED PCE can increase 13,91% bandwidth utilization in Spanish topology, but its computational time increases 49,24% in comparison with two coordinated TEDs per layer.

Acknowledgements

This work has received funding from the EC FP7 Programme ICT STRONGEST under grant agreement no 247674 and the ONE project in contract number INFSO-ICT-258300.

References

- [1] J. E. Gabeiras, et al., "Is multilayer networking feasible?", Optical Switching and Networking, vol. 6, no. 2, pp. 129-140 (April, 2009).
- [2] V. López, et al., "Path computation element in telecom networks: Recent developments and standardization activities," in Proc. ONDM, 2010.
- [3] M. Chamania, et al., "Coordinated Computation of Multi-layer Paths via Inter-layer PCE Communication: Standards, Interoperability and Deployment," in IEEE ICC workshop on Telecommunications: From Research to Standards, Jun 2012.
- [4] H. Zhu, et al., "A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks," IEEE/ACM Trans. , 2003.
- [5] O. Gonzalez de Dios, et al., "Functional validation of the cooperation VNTM and PCE", in Proc. iPOP, Jun 2012

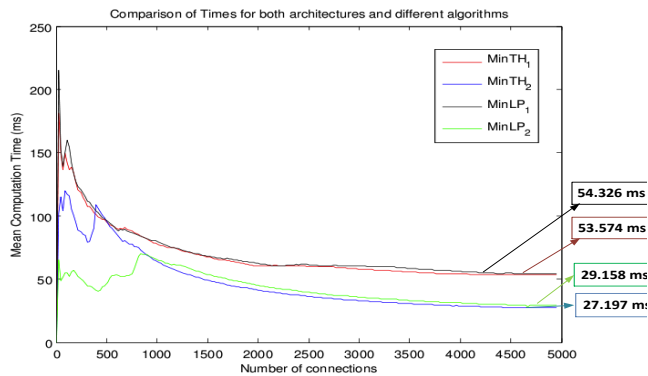


Fig. 5. Mean computation time (ms)

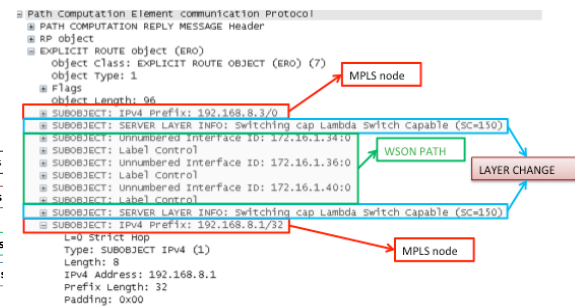


Fig. 6. Wireshark trace of PCEP Reply message

D PRESUPUESTO

El presupuesto de este proyecto se va a desglosar en las siguientes secciones generales:

- 1) Recursos y Ejecución material
- 2) Costes de mano de obra
- 3) Gastos generales y beneficio industrial
- 4) Presupuesto total

1) Ejecución Material:

En primer lugar vamos a desglosar estos costes en tres sub-apartados: costes de software, costes de hardware y costes de material de oficina. La primera tabla desglosa los costes asociados a la partes de software: programas, licencias, etc. Para hacerlo más real se muestra primeramente el coste del uso de cada software durante un año, y a continuación se muestra el tiempo real de uso de dicho software y el gasto que supone adaptado a dicho período de uso:

Gasto Recursos Software			
Programa	Coste	Meses de uso	Coste Real
Windows 7 Enterprise 32 bits	269,0	6	134,5
Ubuntu Server 10.4 64 bits	0,0	12	0,0
Microsoft Office 2010	450,0	4	150,0
Microsoft Visio 2010	725,0	5	302,08
Matlab Student Version	99,0	12	99,0
Eclipse SDK Version 3.7	0,0	12	0,0
Total Gasto en Software			685,58

Tabla C-1: Tabla de gastos en recursos de Software

A continuación se desglosan los gastos de hardware:

Gasto Recursos Hardware	
Dispositivo	Coste
Ordenador de sobre mesa – Modelo Dell Optiplex 745 Intel Core™2 CPU 4400 @ 2200GHz 2.00GHz	450,0
Monitor 17 pulgadas – Dell	120,0
Servidor externo – Dell PowerEdge R310 Intel® Xeon® CPU X3430 @2.40GHz	990,0
Total Gasto en Hardware	1560,00

Tabla C-2: Tabla de gastos en recursos Hardware

Por último los recursos materiales junto con el total de gasto en recursos de las 3 secciones:

Recurso Material y Gasto Total	
Recurso	Coste
Recursos Software	685,58
Recursos Hardware	1560,00
Recursos Materiales de oficina	150,0
Total Gasto en Hardware	2.289,58

Tabla C-3: Tabla de gastos en recursos materiales

2) Costes de Mano de obra:

Para la elaboración del proyecto han sido necesarios los siguientes perfiles:

- 1) Estudiante de Ingeniería superior de Telecomunicación encargado del desarrollo y la implementación necesarias para llevar a cabo las simulaciones y los correspondientes resultados (trabajo técnico). (IST)
- 2) Un Doctor en Ingeniería de Telecomunicación encargado de la dirección del proyecto de la presentación de los resultados obtenidos. (DIT)

Para entender más a fondo estos costes vamos a explicar la planificación detallada de las tareas que conlleva este proyecto, como se ve en la siguiente tabla. Todas las duraciones están expresadas en meses:

Secuencia de Tareas del Proyecto				
Tarea	Descripción	Esfuerzo	Duración	
1) Estudio del estado del arte de los algoritmos de RWA y de rutado multicapa.	En esta primera etapa se comenzó investigando sobre el estado del arte de algoritmos de rutado en redes WSON.	IST - 1 personas-mes.	12	
2) Implementación y desarrollo de algoritmos de rutado WSON en el testbed del CTTC	Desarrollo de los algoritmos del estado del arte en código C++ para aplicarlos al testbed del CTTC.	IST – 1 persona-mes.	1	
3) Desarrollo e implementación de los algoritmos para el testbed de TID	Desarrollo de los mismos algoritmos de rutado y selección de longitud de onda para el testbed de TID.	IST – 1 personas-mes.	2	
4) Simulaciones y obtención de resultados	Simulaciones en ambos testbeds para obtener los resultados de rendimiento en términos de probabilidad de bloqueo.	IST – 1 personas-mes	2	
5) Implementación de los algoritmos de rutado en redes multicapa.	Desarrollo de los algoritmos del estado del arte para el rutado de paquetes en entornos multicapa.	IST – 1 peronas/mes.	4	
6) Documentación y difusión.	Escritura del presente proyecto final de carrera, así como elaboración de las publicaciones presentadas en este documento.	IST – 1 personas-mes.	4	
7) Gestión/Dirección del Proyecto	Gestión y evaluación de todas las tareas elaboradas en el Proyecto	DIT – 0,15 personas-mes.	18	

Tabla C-4: Tabla de planificación del proyecto desglosado en tareas

En total el proyecto ha tenido una duración de 12 meses.

Basándonos en el desglose de tareas realizado previamente podemos explicar ahora los costes asociados a la mano de obra en el proyecto. La estimación de costes se hace en base a los siguientes datos:

- Cotizaciones según el Régimen General de la Seguridad Social. Tanto el Ingeniero como el Doctor pertenecen ambos al grupo 1.
- Jornada laboral de 8 horas/día y 21 días laborables/mes.

Con estos datos se obtiene la siguiente tabla:

Costes Salariales		
Concepto	Grupo 1	
	IST	DIT
Base Cotizable máxima Anual	41.108,40	41.108,40
Contingencias comunes (23,6%)	9.701,58	9.701,58
Desempleo (5,5%), F.G.S. (0,20%) y Formación profesional (0,6%). Porcentajes IST Sobre salario bruto	2.268	3.185,90
Coste de la seguridad social	11.969,58	16.313,18
Salario bruto anual	36.000,00	42.000,00
Coste Salarial anual	47.969,58	58.313,18
Coste Salarial por hora	25,95	28,92
Número de horas	880	20,53
Coste Total	22.836	593,72

Tabla C-5: Tabla de costes salariales

Coste Mano de Obra	
Concepto	Coste
Ingeniero Superior de Telecomunicación	22.836
Doctor Ingeniero de Telecomunicación	593,72
Coste Total	23.429,72

Tabla C-6: Tabla de costes de mano de obra

3) Gastos Generales y Beneficio Industrial:

Bajo gastos generales se encuentran todos aquellos gastos derivados del uso de las instalaciones, cargas fiduciarias, amortizaciones, gastos fiscales etc...

Presupuesto de Ejecución por contrata	
Concepto	Coste
Presupuesto de ejecución material	
Costes de mano de obra	23.429,72
Gastos en Recursos materiales	2.289,58
Sobtotal	25.719,3
Gastos generales (13% del P.E.M)	3.343,50
Beneficio Industrial (6% del P.E.M)	1.543,15
Total presupuesto de ejecución por contrata	30606,23

Tabla C-7: Tabla de presupuesto de ejecución por contrata

4) **Presupuesto Total:**

Presupuesto Total	
Concepto	Coste
Presupuesto de Ejecución por contrata	30.606,23
I.V.A. (21%)	6.427,30
TOTAL	37.033,54

Tabla C-8: Tabla de presupuesto Total

Madrid, 10 de Julio de 2013

Madrid, Julio de 2013

El Ingeniero Jefe de Proyecto

Fdo.: Sergio Martínez de Tejada Ayuso
Ingeniero Superior de Telecomunicación

PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización de este proyecto: *Implementación y evaluación de algoritmos de cálculo de ruta y selección de longitud de onda para un PCE multicapa*. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación.

5. Aparte del Ingeniero Doctor, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Doctor de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Doctor de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partidaalzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Doctor que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.